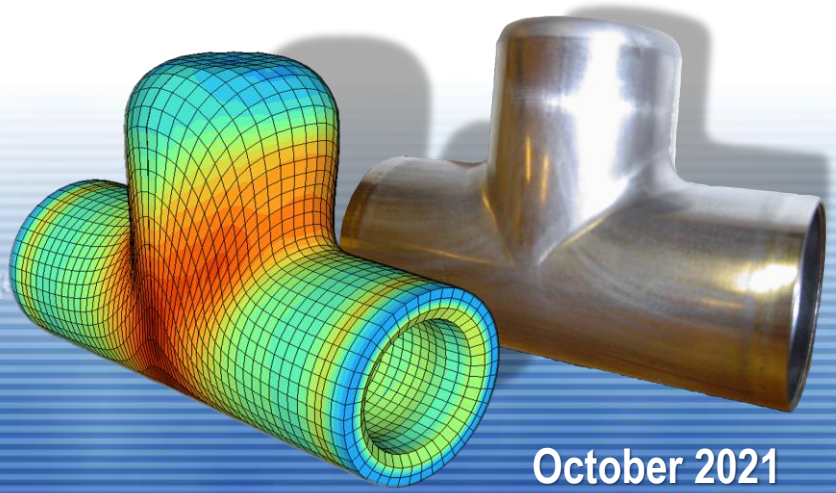
Metafor Tutorial for the advanced solid mechanics project

Cédric Laruelle, Romain BOMAN

ULiège – Dept of Aerospace and Mechanical Engineering



October 2021

Outline

1. What is Metafor?
2. How to install Metafor?
3. How to run an existing test?
4. How to modify an existing test?
5. FAQ

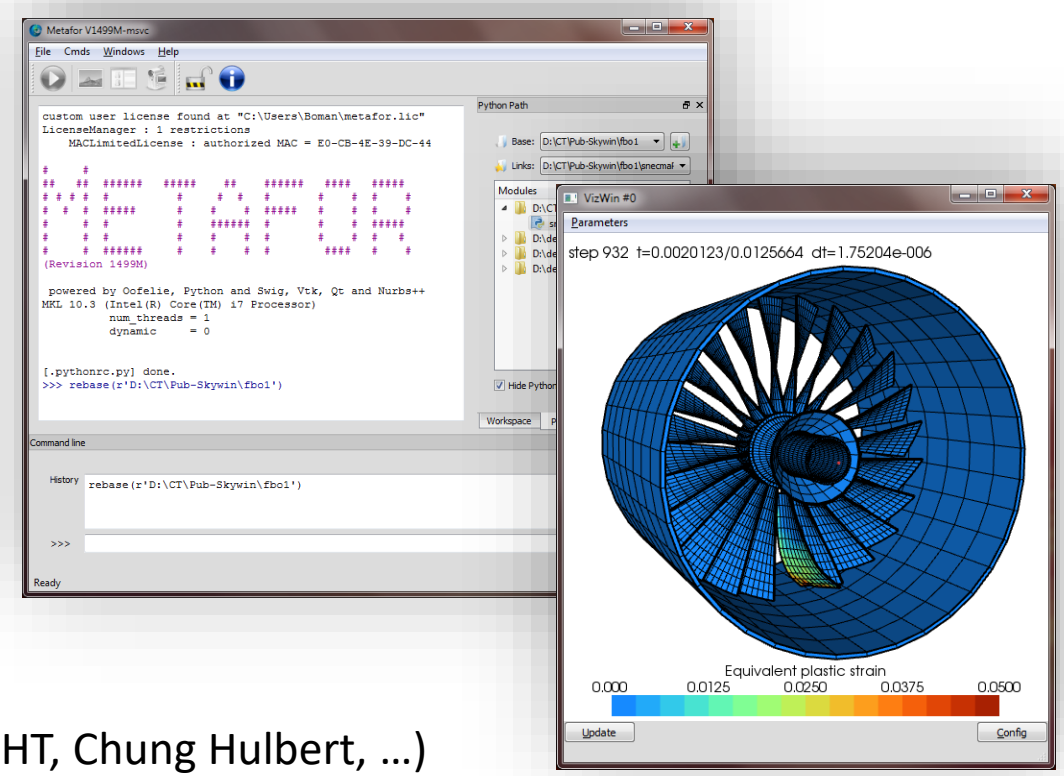
Outline

1. **What is Metafor?**
2. How to install Metafor?
3. How to run an existing test?
4. How to modify an existing test?
5. FAQ

What is Metafor?

Metafor

an object-oriented Finite Element code for the simulation of solids submitted to large deformations



Main coding languages

- C++: algorithms
- Python: input files

Main features

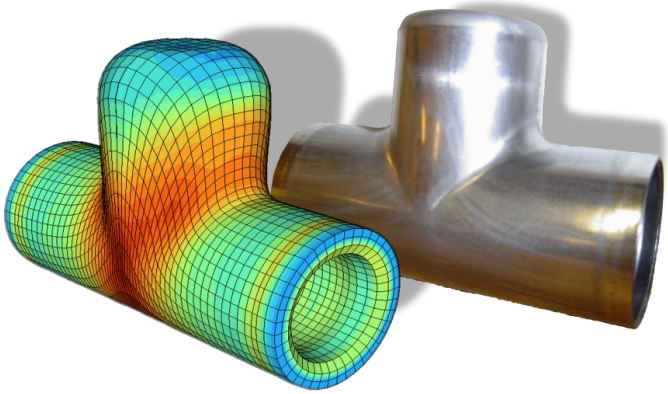
- 2D/3D elements (large strains).
- Implicit/explicit time integration (HHT, Chung Hulbert, …)
- Thermomechanical coupling (staggered or fully coupled schemes).
- Frictional contact between deformable bodies or analytical surfaces.
- Arbitrary Lagrangian Eulerian formalism.
- Meshing and remeshing procedures.
- Large set of constitutive laws (thermo-elasto-visco-plastic, damage, …)
- Crack propagation (erosion method).

python powered

What is Metafor?

Metafor…

- … is a numerical implementation of the algorithms described in course MECA-0464.
- … includes all the PhD theses, Final Year Projects and research projects performed at Prof Ponthot's laboratory.
- … is not a commercial software (less robust but more modular).

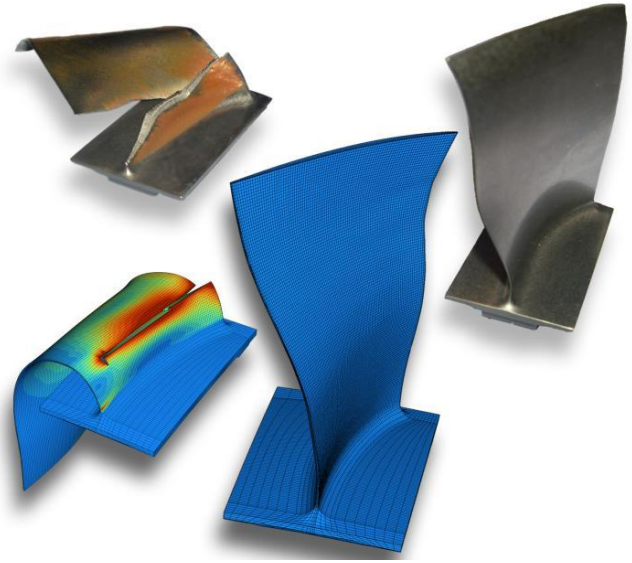


Hydroforming of a tube

Typical applications

- Metal forming processes (roll forming, deep drawing, roller levelling, hydroforming, etc.)
- Crash, impact (fan blade out, shock absorbers, etc.)
- Biomechanics (brain shift, orthodontics, etc.)
- Fluid/structure interaction

Website (in English/French)
http://metafor.ltas.ulg.ac.be/

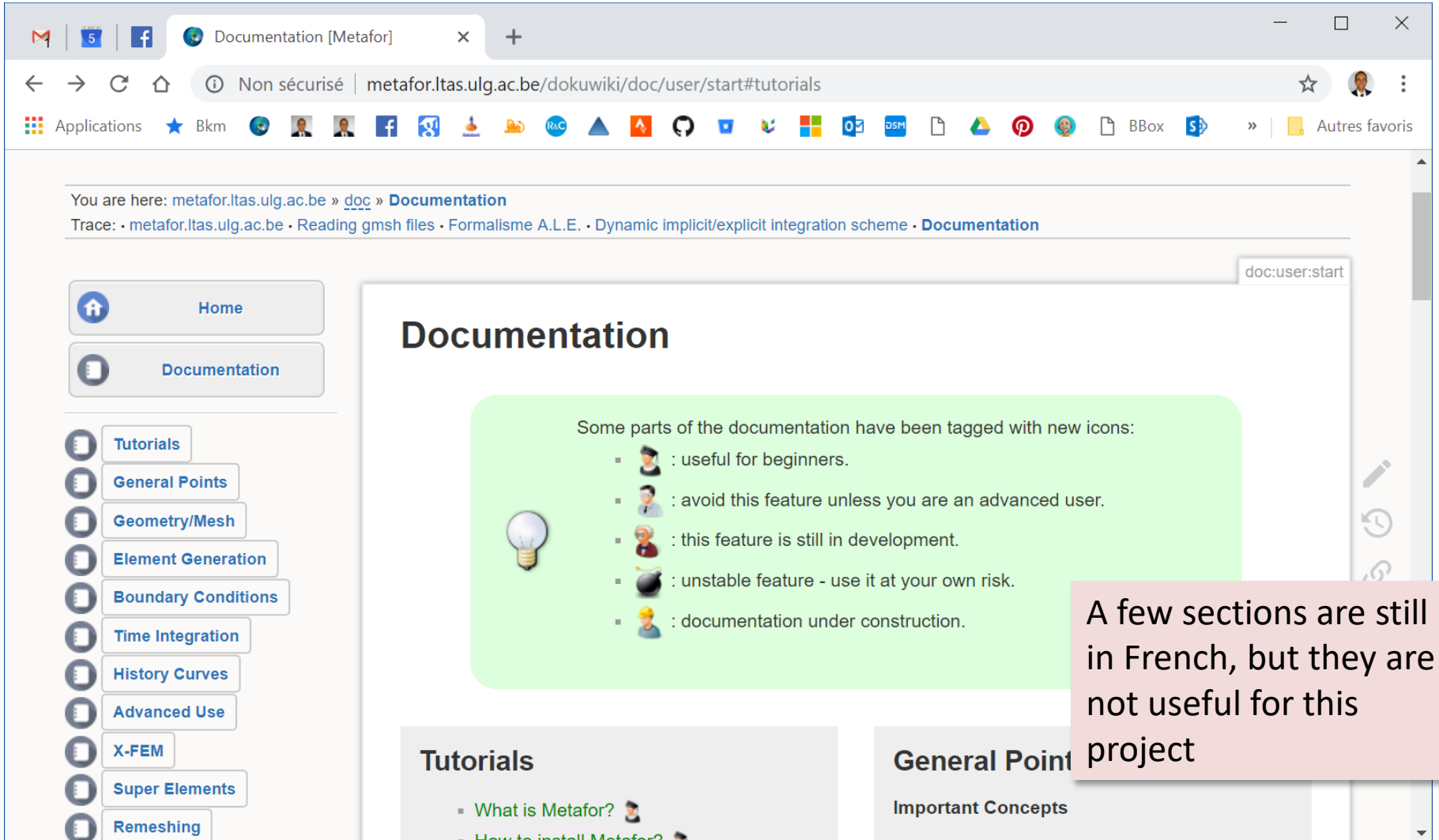


Impact of a aeroengine blade onto a casing

What is Metafor?

Documentation website

http://metafor.ltas.ulg.ac.be/dokuwiki/doc/user/start



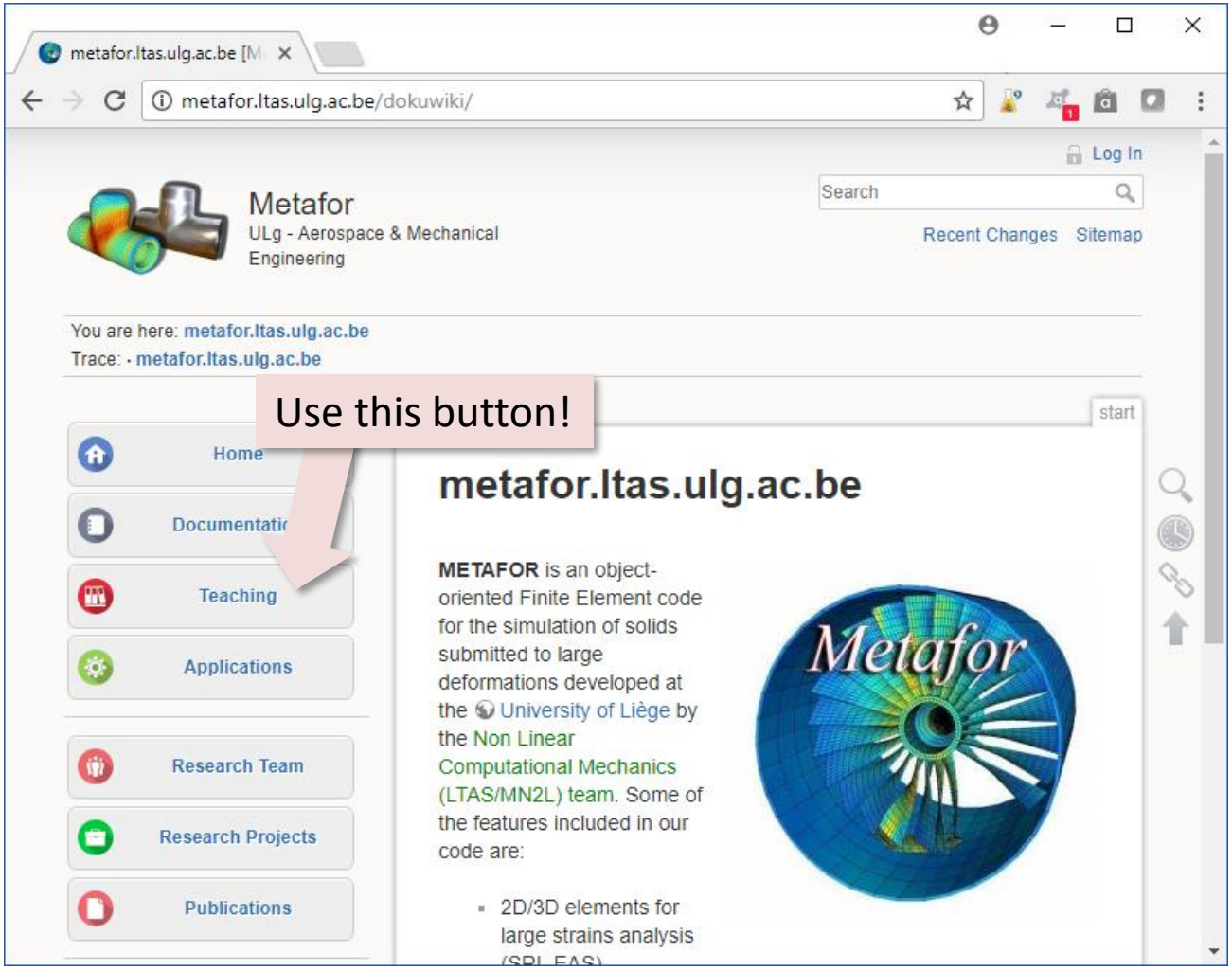
A few sections are still in French, but they are not useful for this project

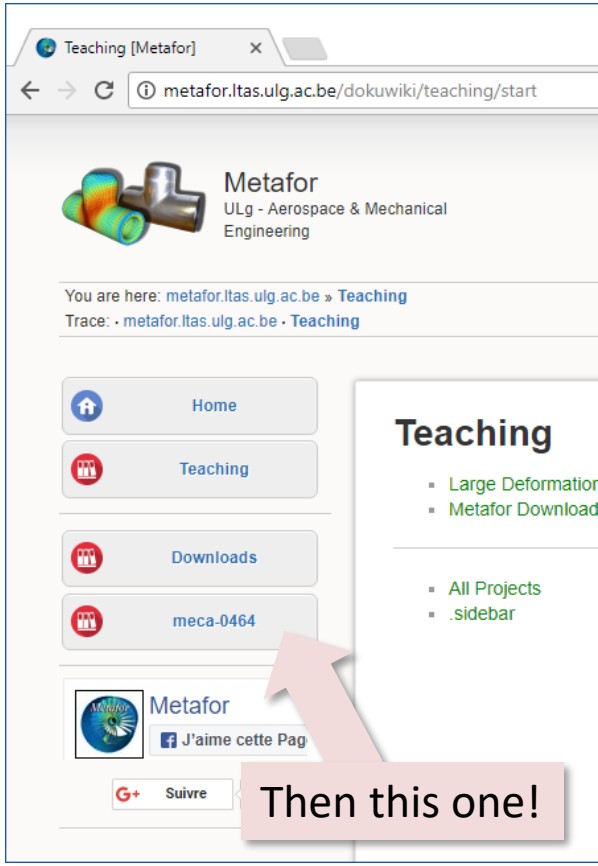
Outline

1. What is Metafor?
2. **How to install Metafor?**
3. How to run an existing test?
4. How to modify an existing test?
5. FAQ

How to install Metafor?

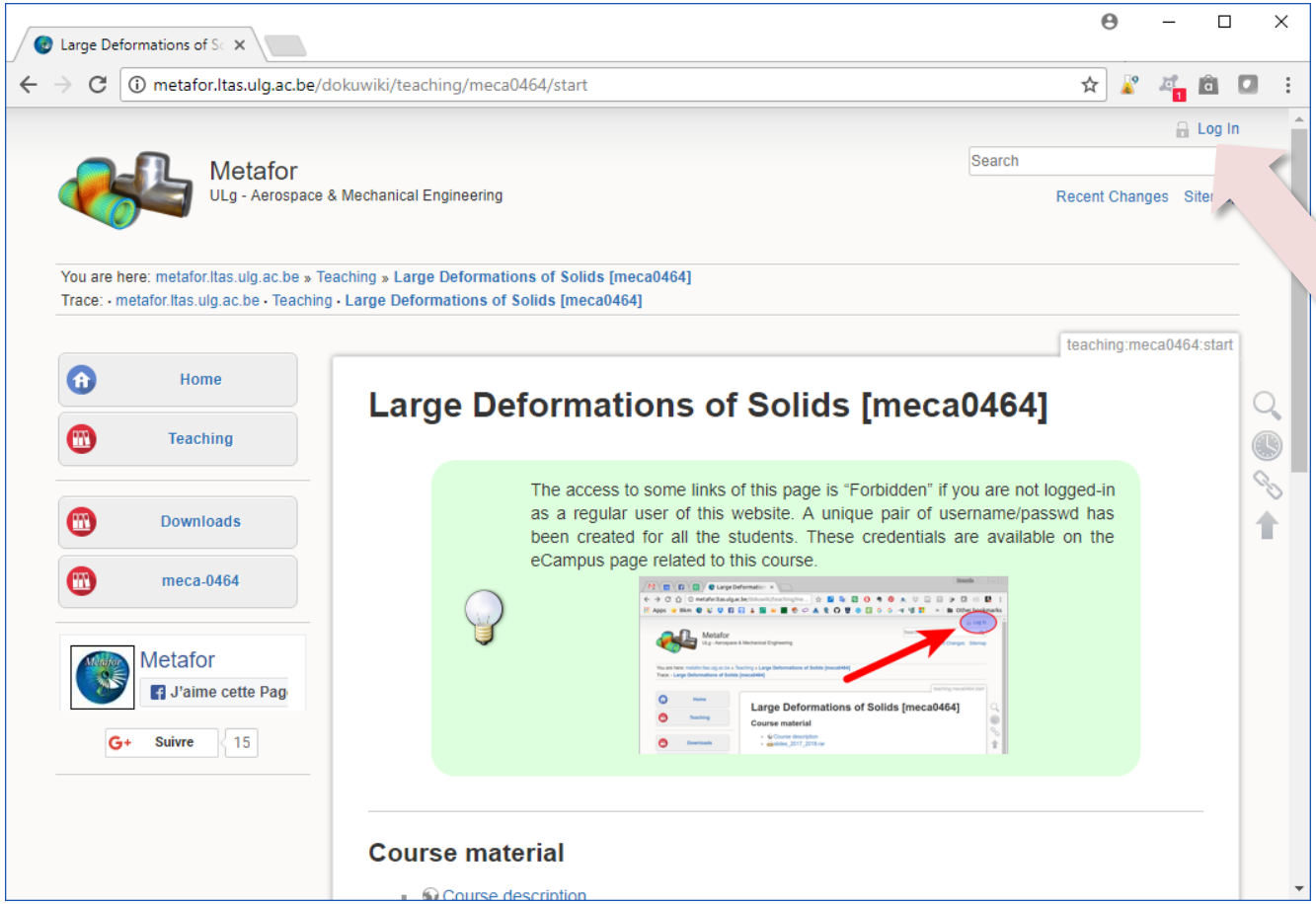
Installers for Windows and other platforms are available on metafor.ltas.ulg.ac.be website (eCampus website is slow, very difficult to manage and sometimes broken)





How to install Metafor?

You are now on the page of the “large deformations” course!



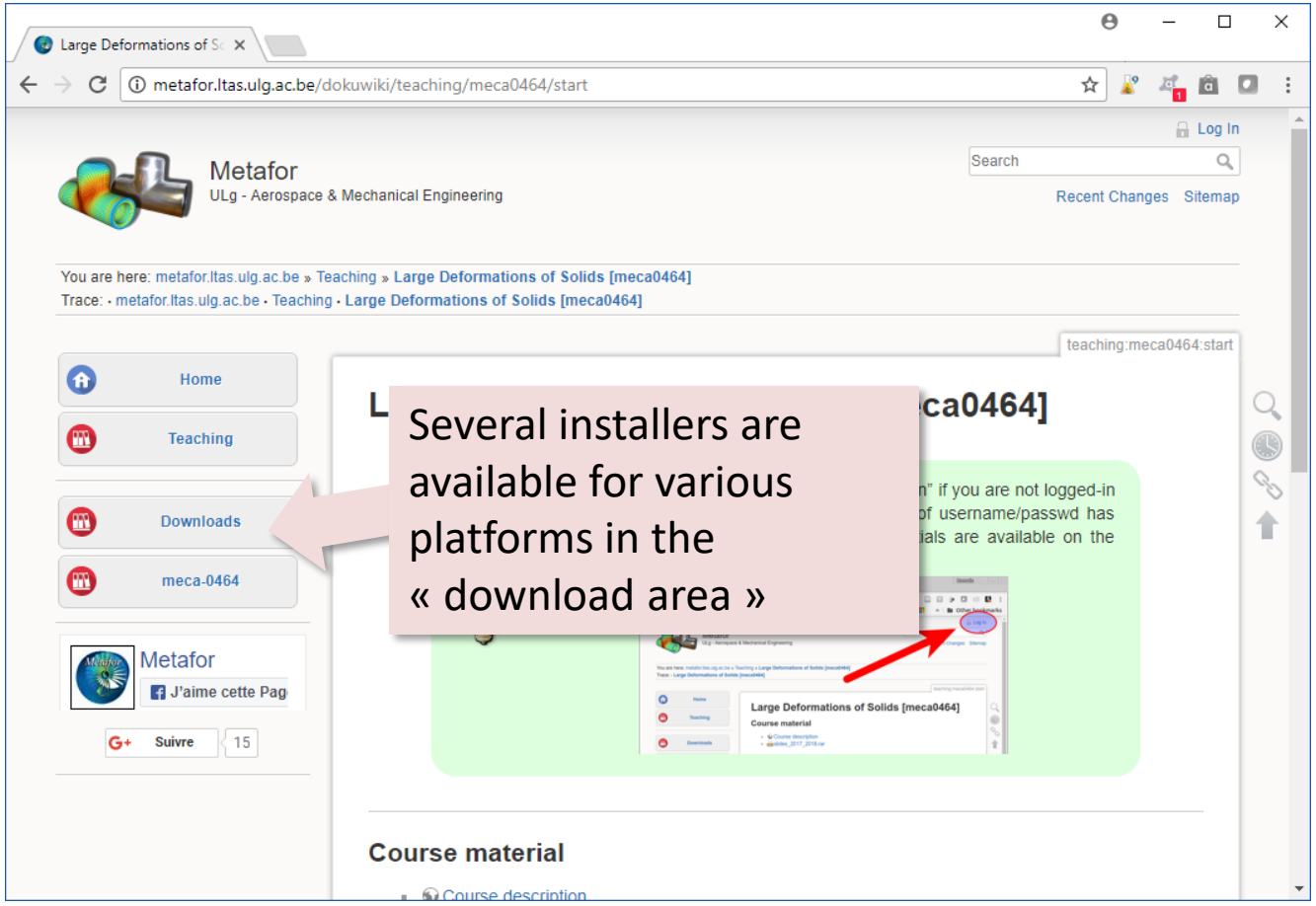
Log in using the following credentials:

user: student
passwd: metafor

so that you are allowed to download files from this page

Note:
The link to the website and the username/passwd is also available on the eCampus website

How to install Metafor?



Windows 10 x64

Old Windows
(or Windows x86)

Mac OS
Sierra 10.13.6

ubuntu
18.04 LTS

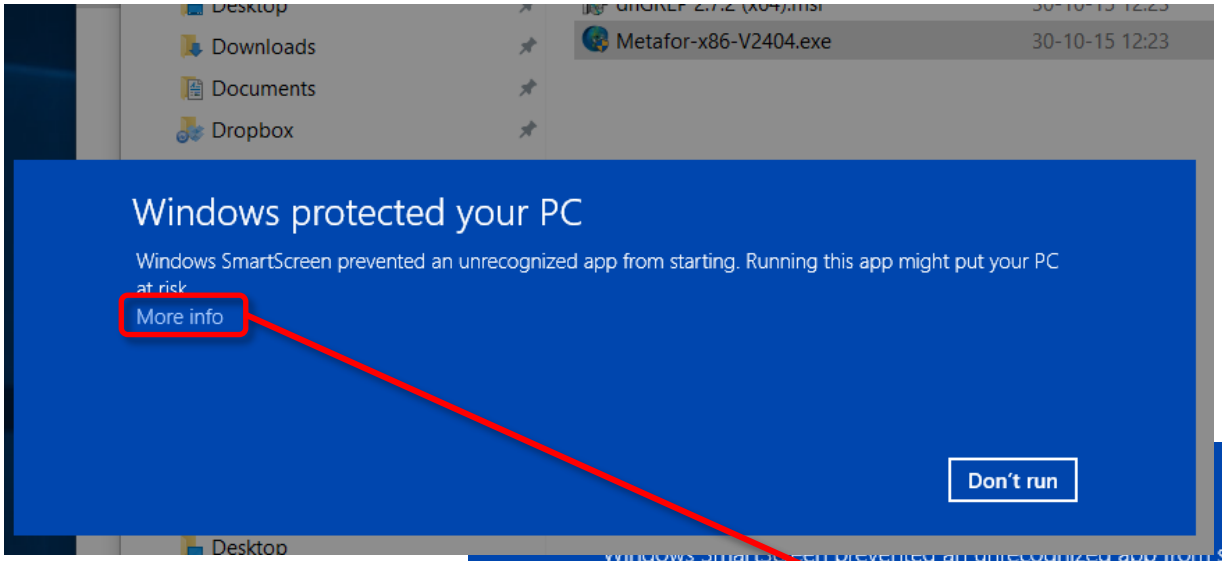
Still having problems?

- ask for help...

How to install Metafor?

Windows Procedure: software installation

1. Run the installer and follow the instructions.
2. If windows says it has protected your PC, click on « more info », then « run anyway »
3. Install the Visual 2015 C++ runtime libraries if they are not present on your system.
4. You can cancel the « External program configurator » (close the window).



These dialog boxes may appear or not, depending on your version of Windows and the fact that you have downloaded the installer from the internet.

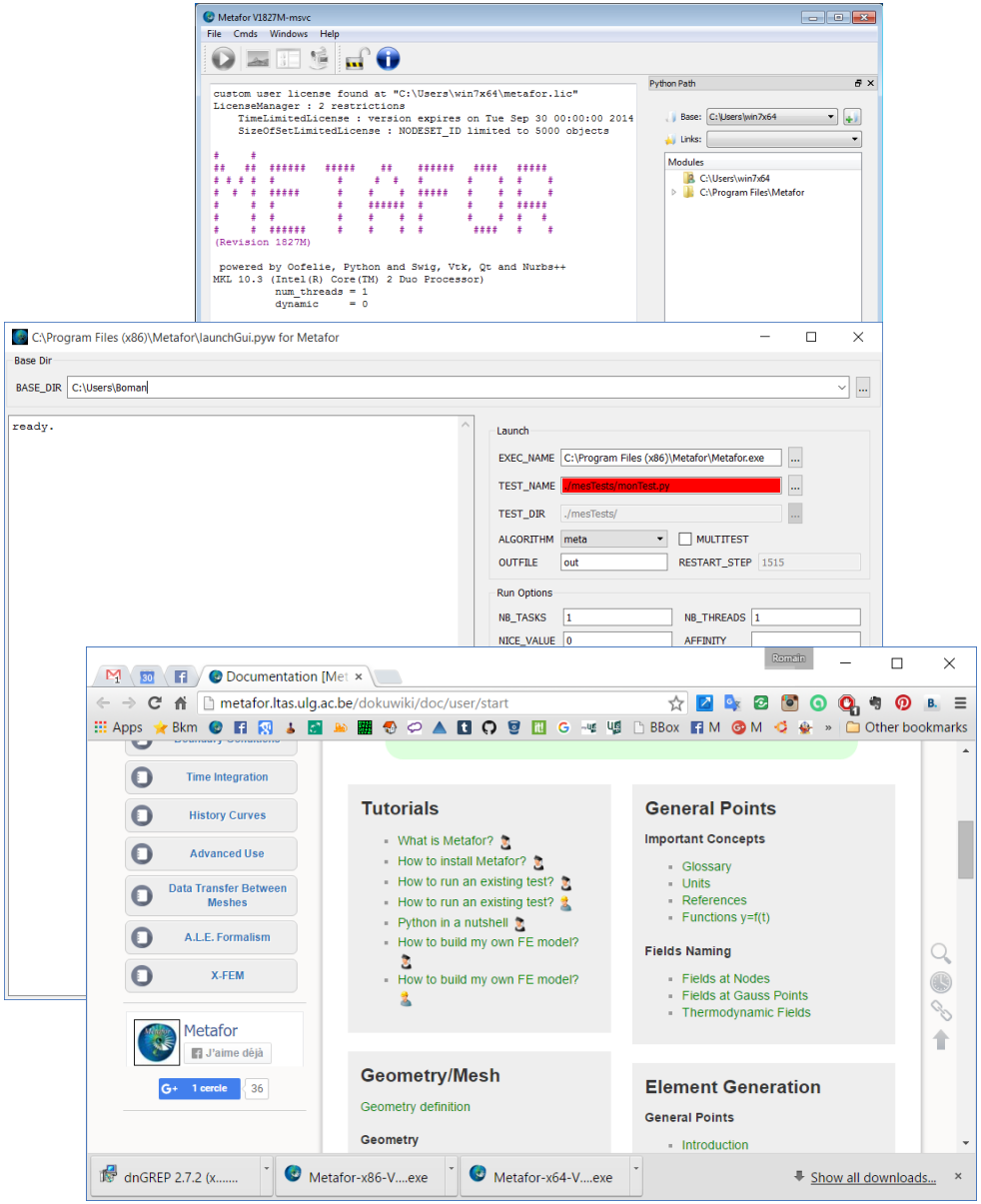
How to install Metafor?

Desktop links:

Run the GUI* of the program

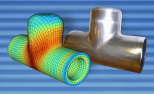
Configuration of advanced runs
(not useful here)

Frozen version of the documentation
(prefer the online version)



* GUI = Graphical User Interface.
Metafor is also available through the command line for long calculations on HPC clusters.

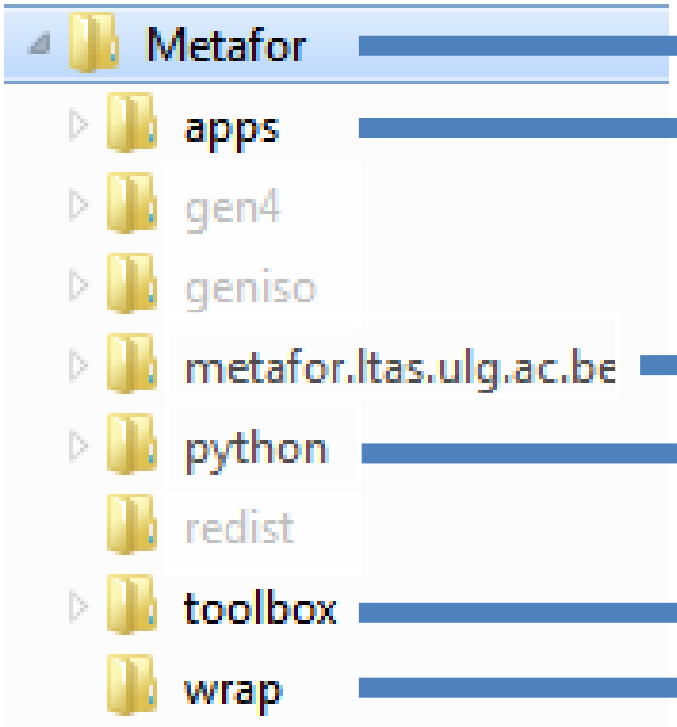
How to install Metafor?



Where is Metafor?

Default location: `C:\Program Files\MetaforVxxxx`
or `C:\Program Files (x86)\MetaforVxxxx`

What has been installed?



Binary files (*.exe, *.dll, etc.)

Lots of examples (input files)

Offline documentation (html files)

Local installation of python

Python toolbox of Metafor

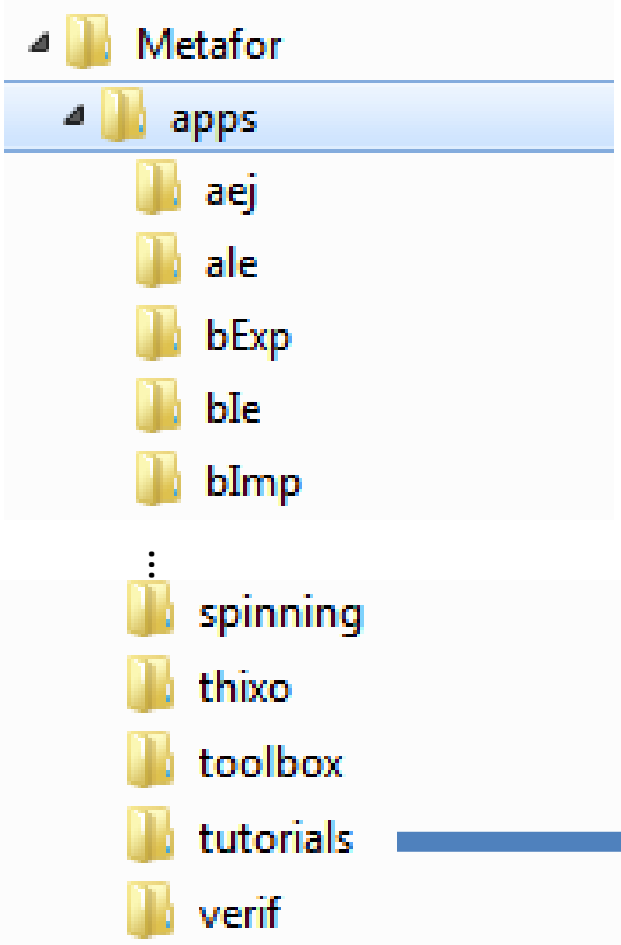
Python/C++ interface

Outline

1. What is Metafor?
2. How to install Metafor?
3. **How to run an existing test?**
4. How to modify an existing test?
5. FAQ

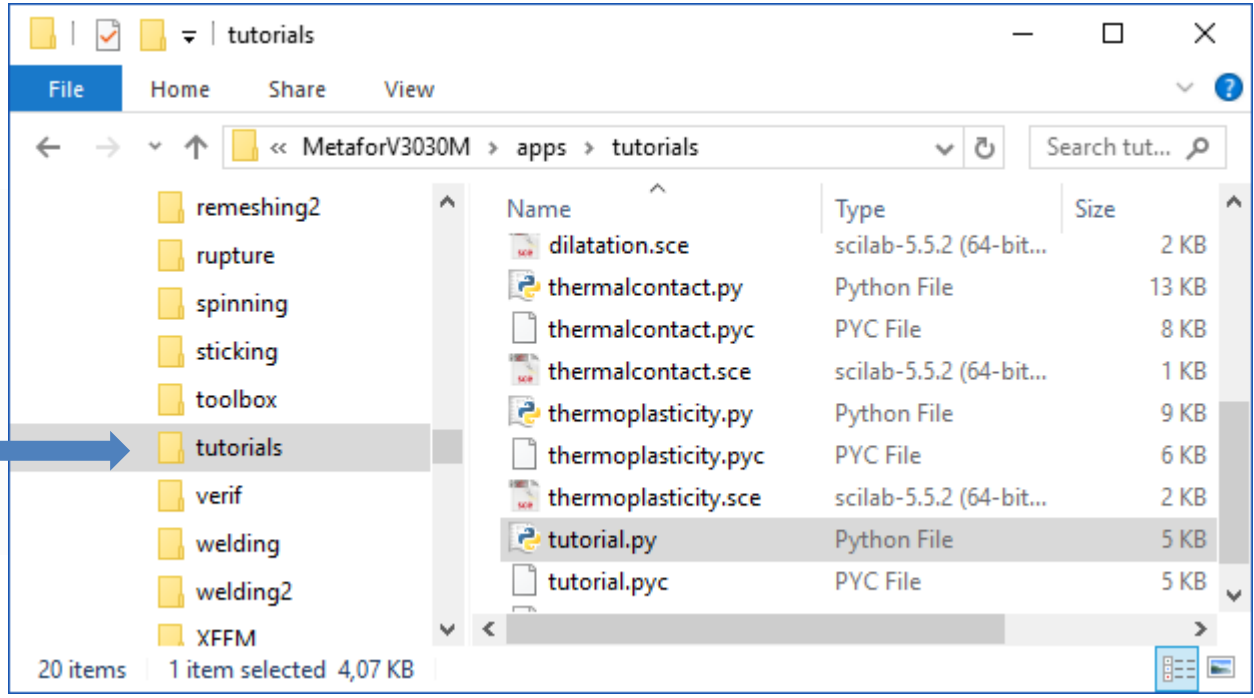
How to run an existing test?

In Metafor: 1 Finite Element (FE) model = 1 (or more) python module(s)



Example: "`tutorial.py`"

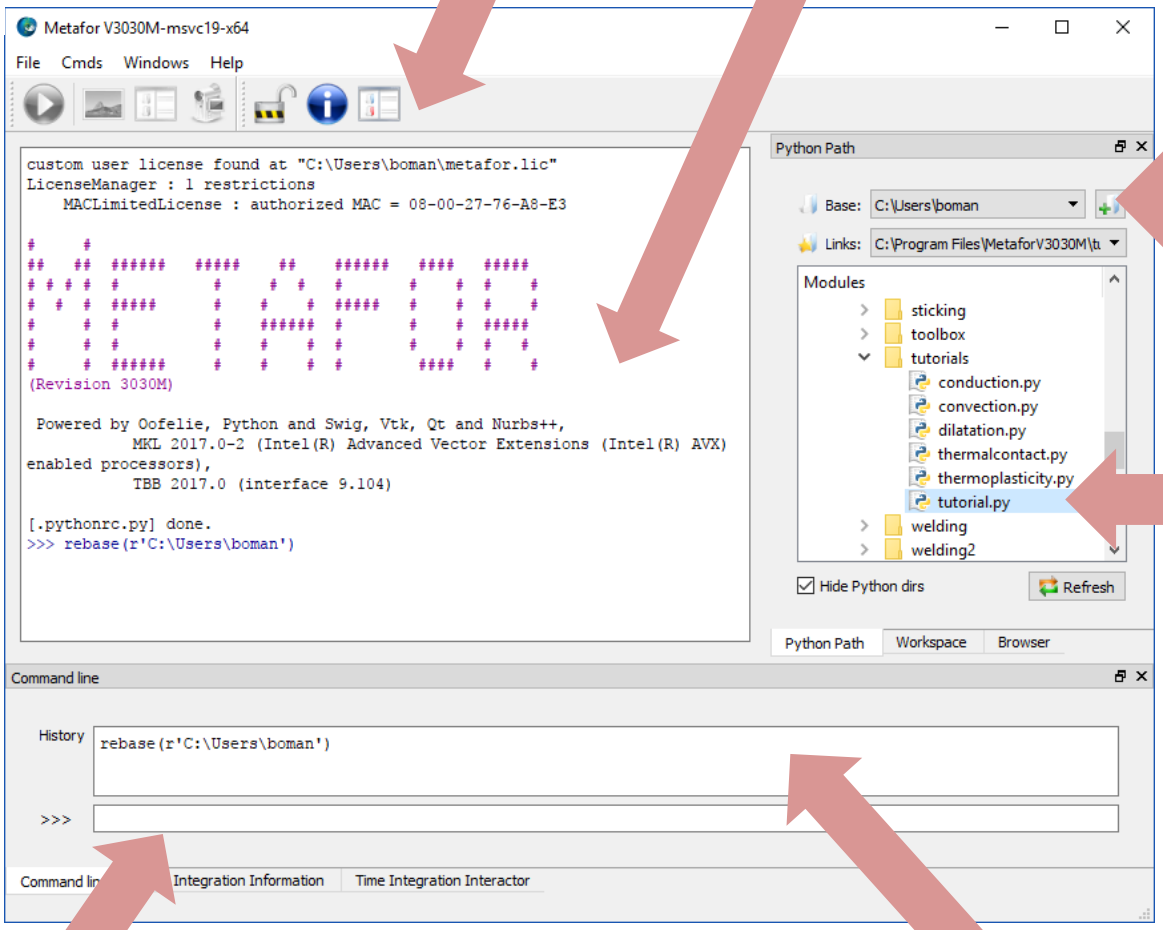
File: `MetaforVxxx\apps\tutorials\tutorial.py`
Module: `apps.tutorials.tutorial`



How to run an existing test?

Toolbar

Output window



"rebase" button (adds any folder to the python path)

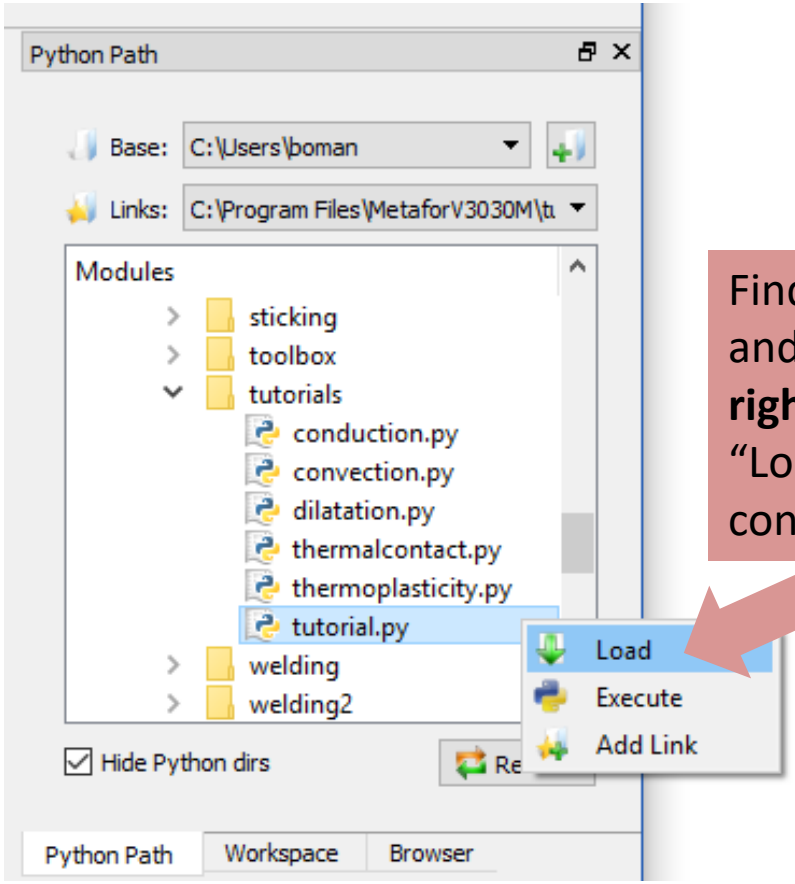
Python path (your module must appear here in order to be run)

Python command line

Command history (use arrows ↑↓)

How to run an existing test?

How to run `apps.tutorials.tutorial` ***?***



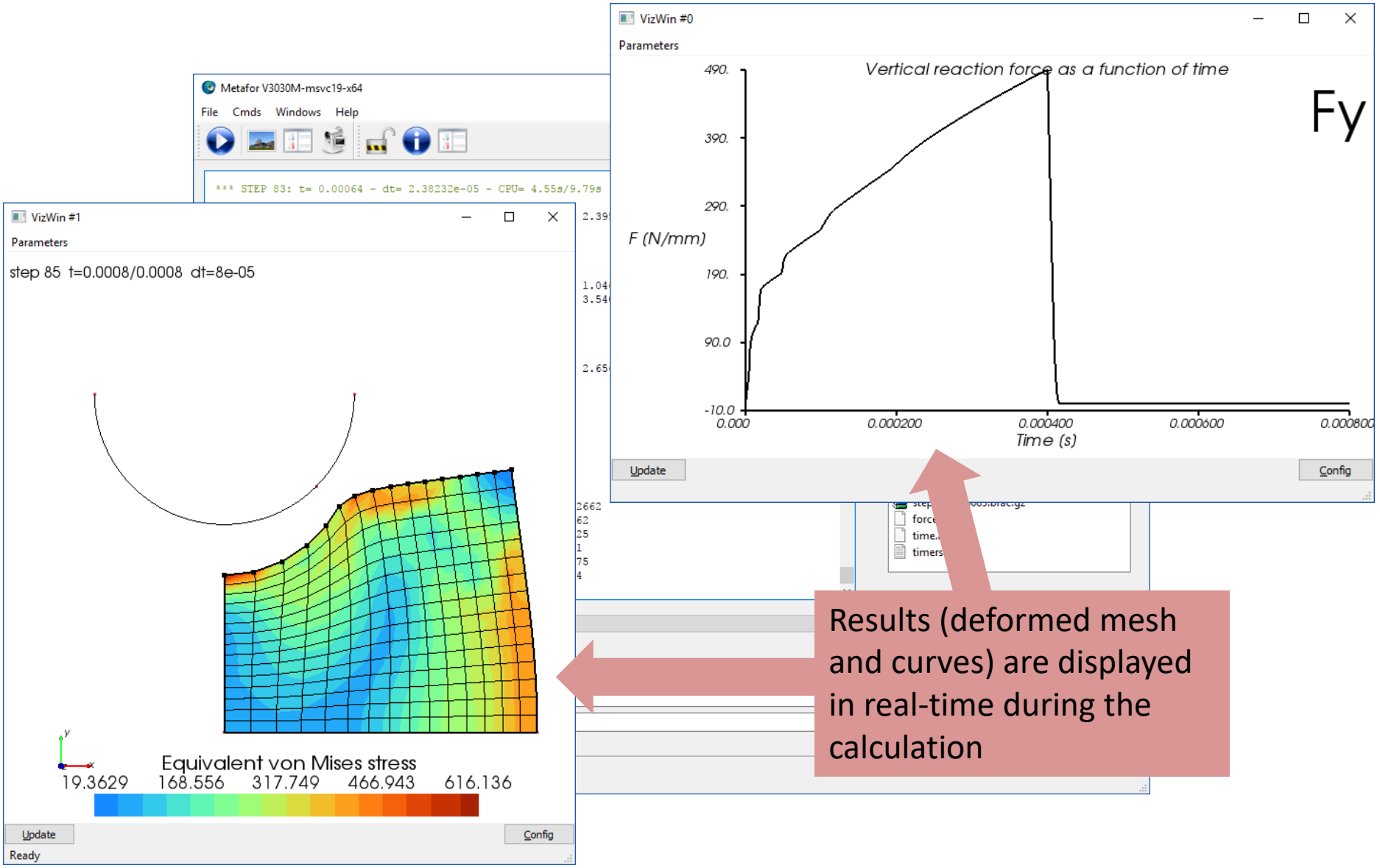
Find the module and **right-click** on "Load" in the context menu

then

Click on the "Play" button

the simulation should start…

How to run an existing test?

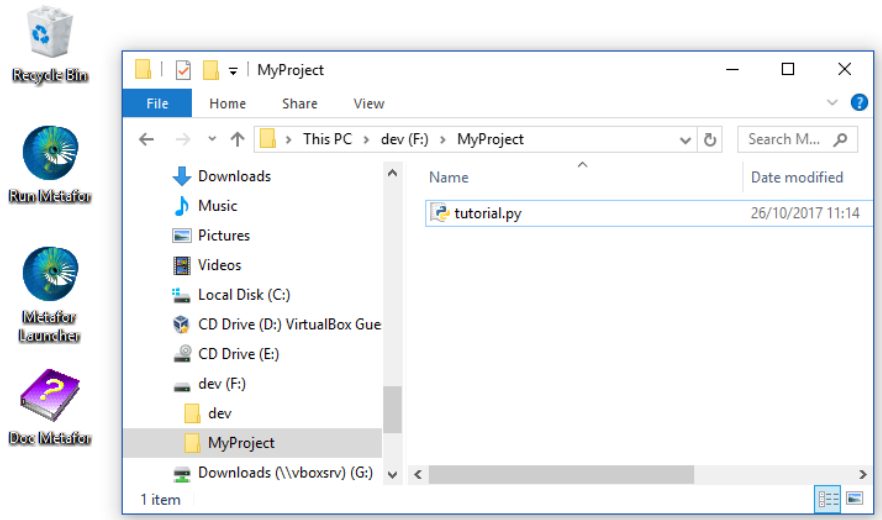


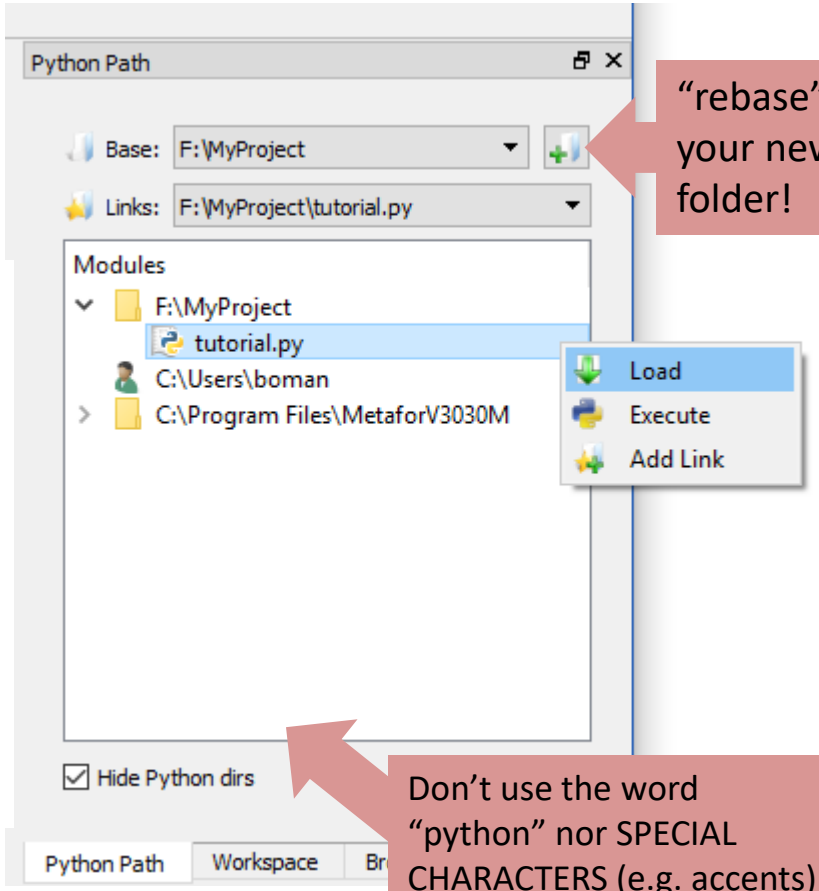
How to run an existing test?

How to run `tutorial` *from another location?*

(because you do not want to work in `c:\Program Files` !)

- Create an empty folder somewhere.
- Copy `tutorial.py` to that folder.
- “Rebase” to that particular folder.





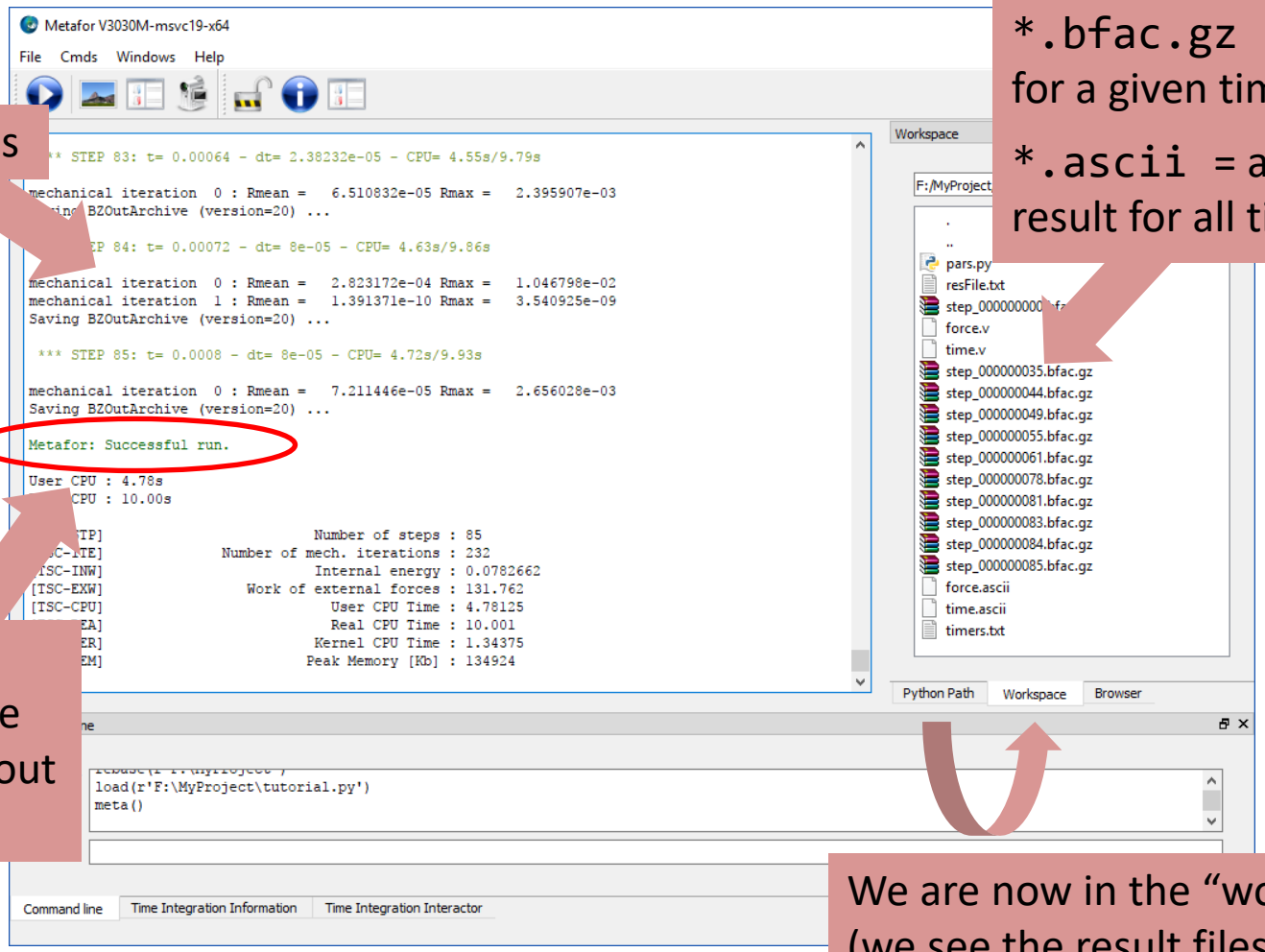
See also:

- Youtube video
- Online documentation (the presented test is `apps.ale.angleReZoner`)

How to run an existing test?

The main window now looks like

Steps/iterations



Results:

`*.bfac.gz` = all results for a given time step

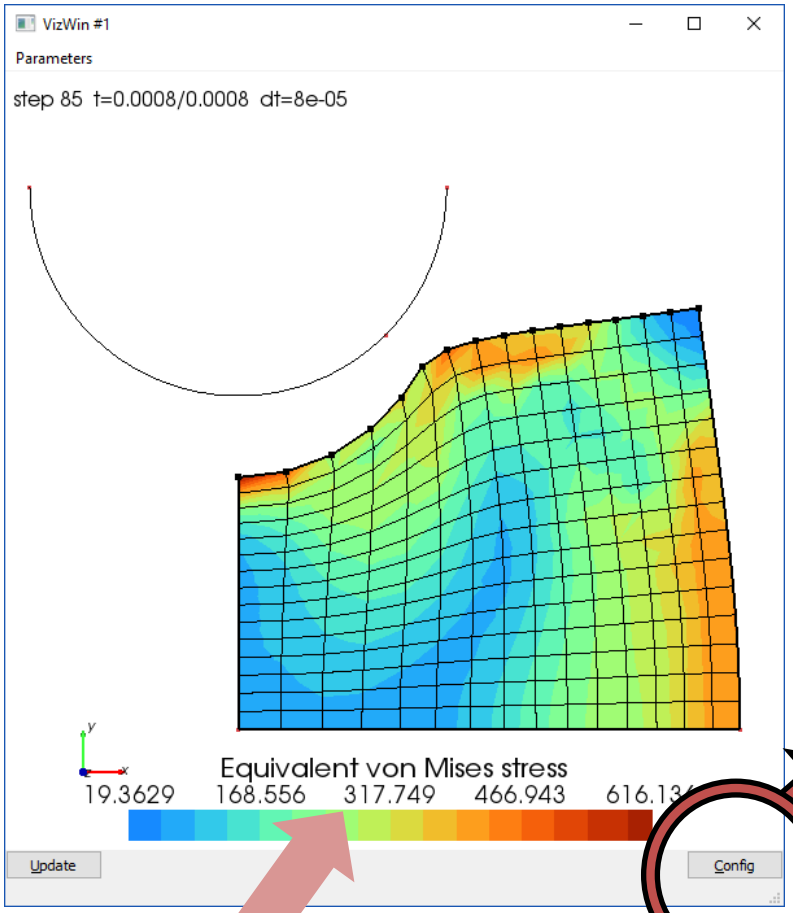
`*.ascii` = a selected result for all time steps

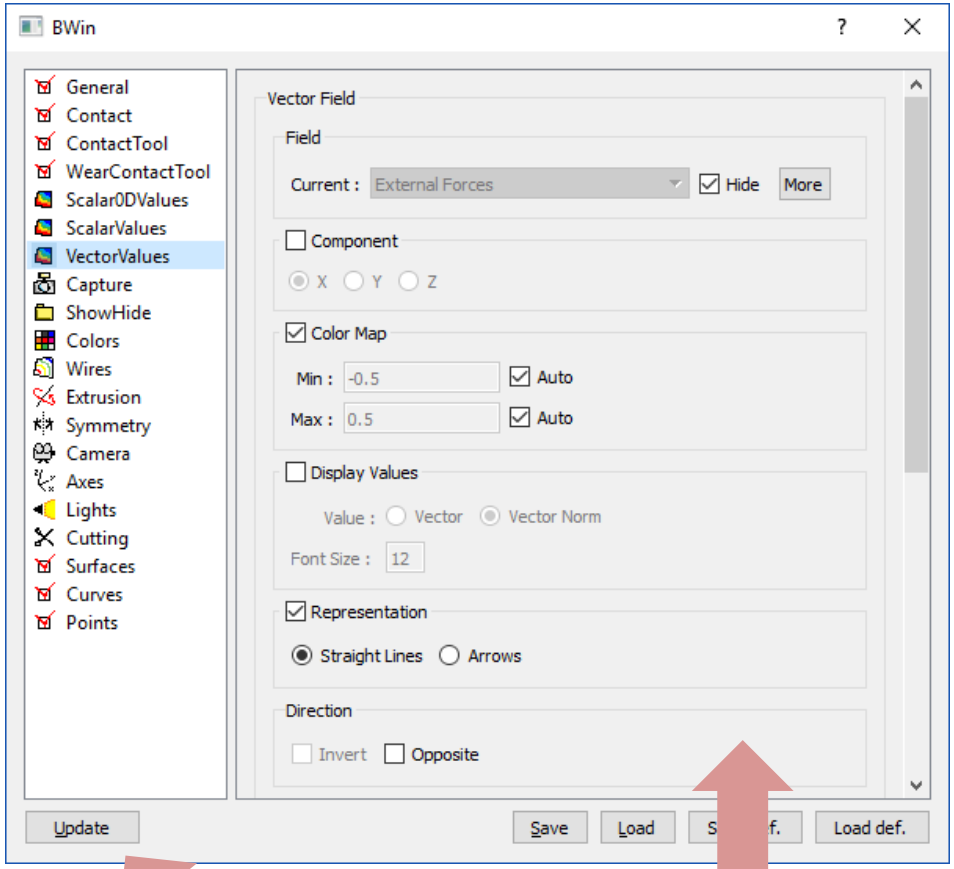
The simulation has reached the final time without any problem!

We are now in the “workspace” (we see the result files instead of the python path)

How to run an existing test?

3D view window:





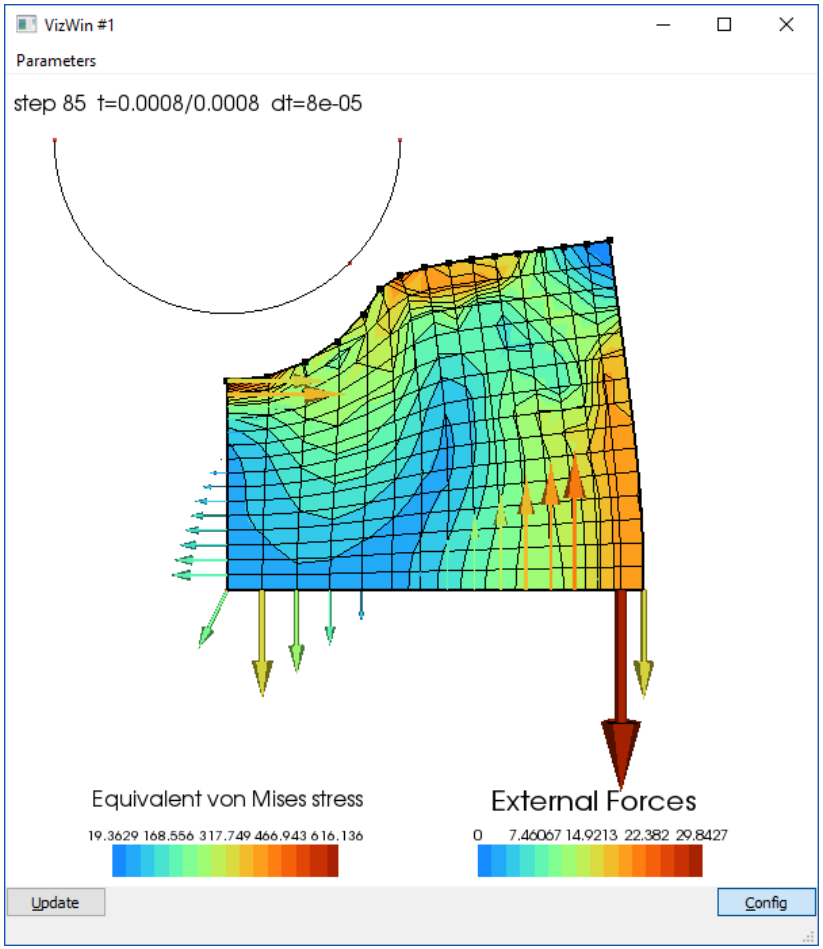
CLICK HERE!

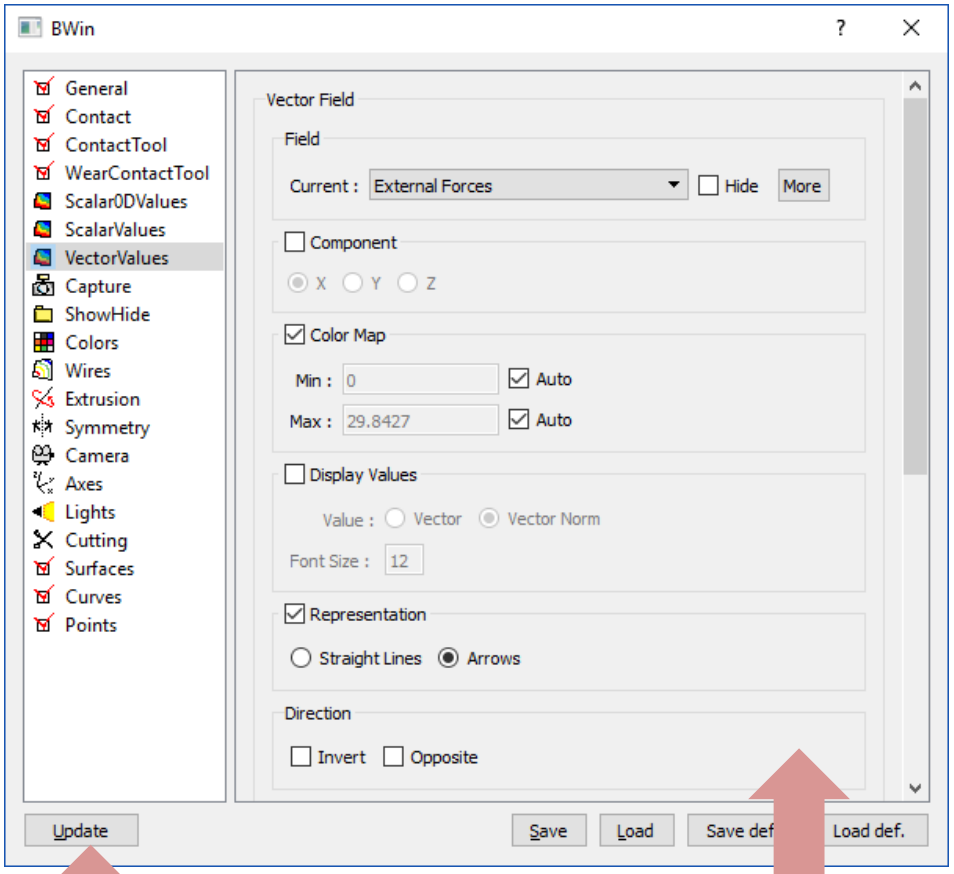
2D / 3D visualization of the model (use the 3 mouse buttons to translate, rotate or zoom)

Configuration of the 3D display (click on the “update” button to apply changes!)

How to run an existing test?

3D view window:

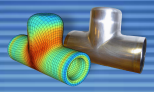




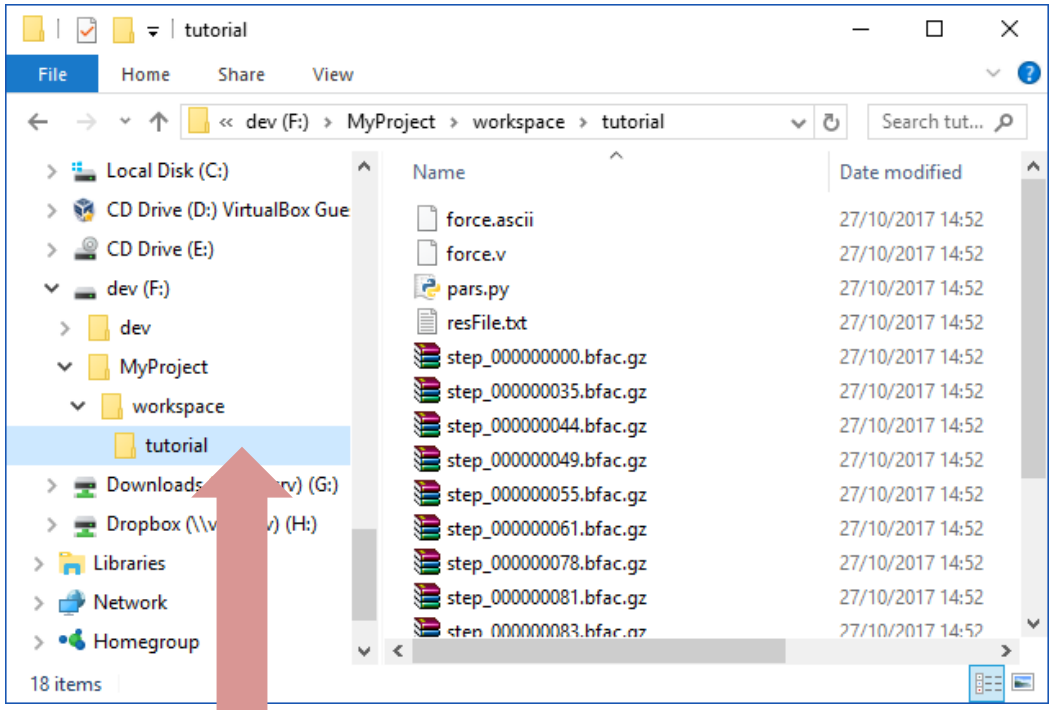
Don’t forget to click on that button.

Lots of (too many?) options are available…
Try them to see what is possible.

How to run an existing test?



On your disk…



`workspace/tutorial` has been created

.bfac.gz files:
1 file = all the results (nodal and Gauss-point values) at 1 time step

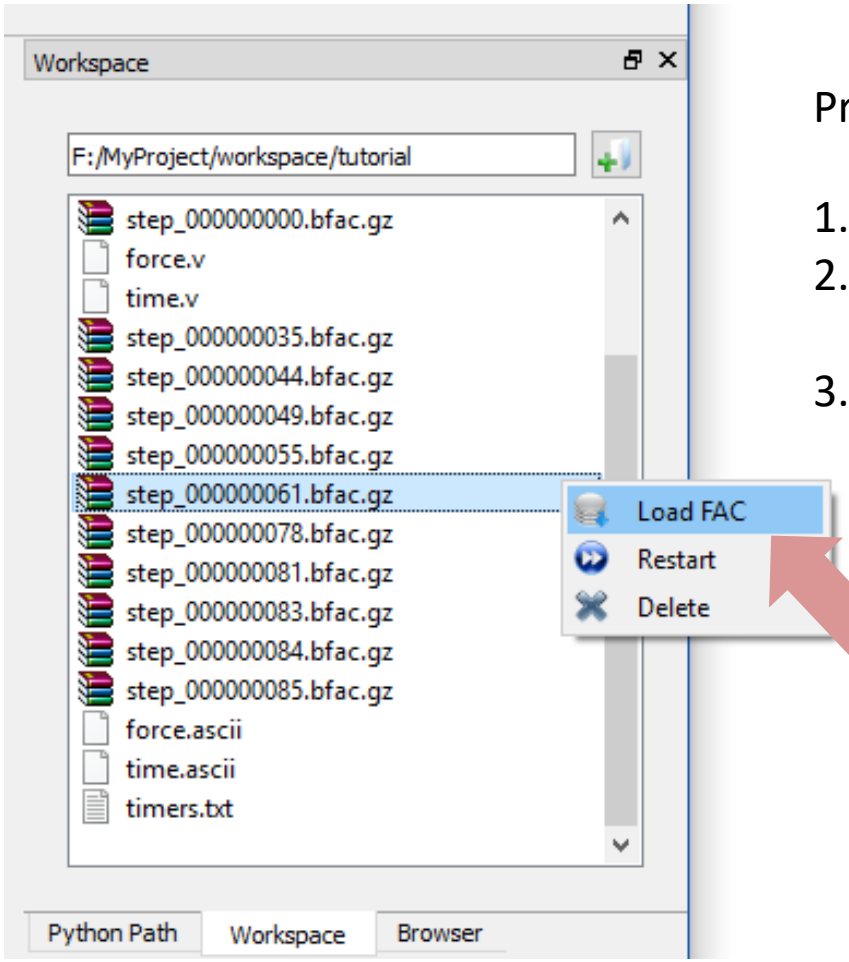
.ascii files:
1 file = 1 vector containing a particular result thoughout the simulation (size = number of time steps)

resfile.txt:
console output during the simulation (steps, iterations, residuals, etc.)

- Here, 11 steps have been saved to disk (there are 11 files with extension `.bfac.gz`)
- The time discretisation (`time.ascii`) and the value of a force somewhere on the solid (`force.ascii`) has been saved

How to run an existing test?

How to display previously-computed results?



Procedure for archives (.bfac.gz files)

1. Start Metafor
2. Load your module (as if you were restarting the simulation)
3. In the workspace, load a `.bfac.gz` using the "load FAC" command in the context menu (right click).

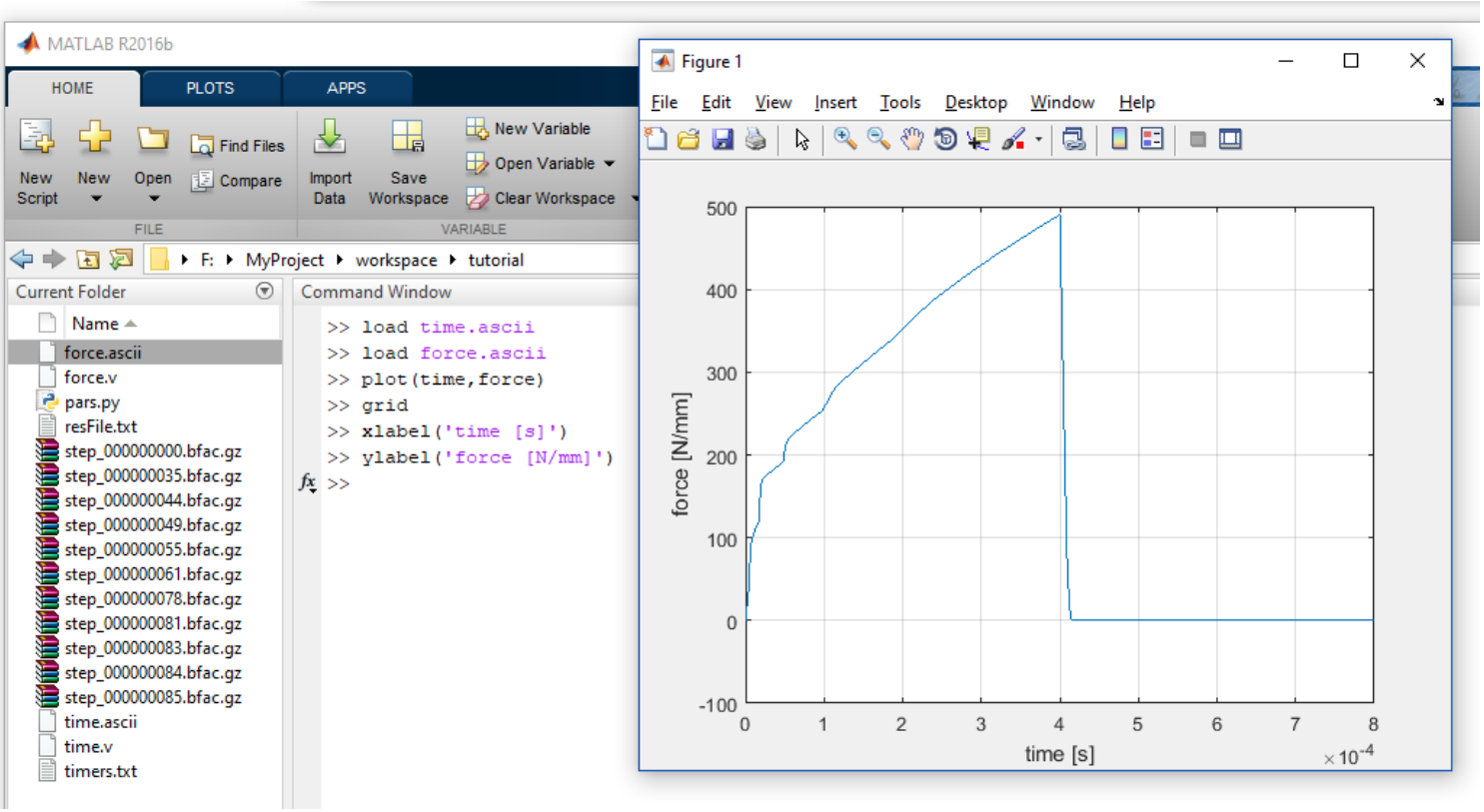
Load time step #61 of `tutorial` into memory and display the mesh and the results at that particular time.

How to run an existing test?

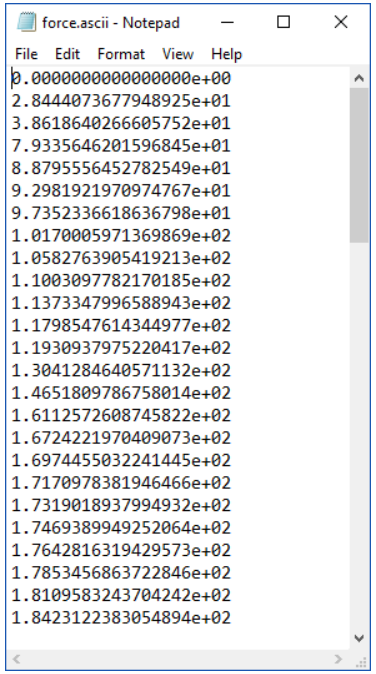
How to display previously-computed results?

Procedure for history curves (.ascii files)

1. Start Matlab
2. Load the files using the “load” command
3. Plot the values



History curves are simple ASCII files!



MS Excel/gnuplot/Scilab/matplotlib can also be used…

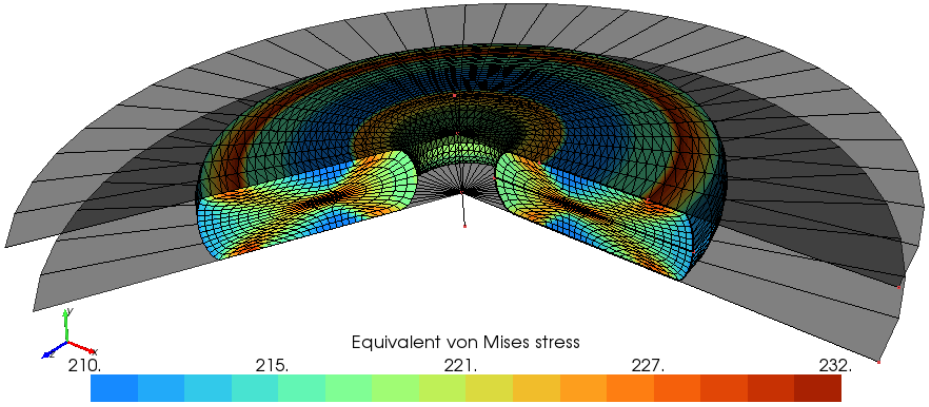
How to run an existing test?

Final Remarks

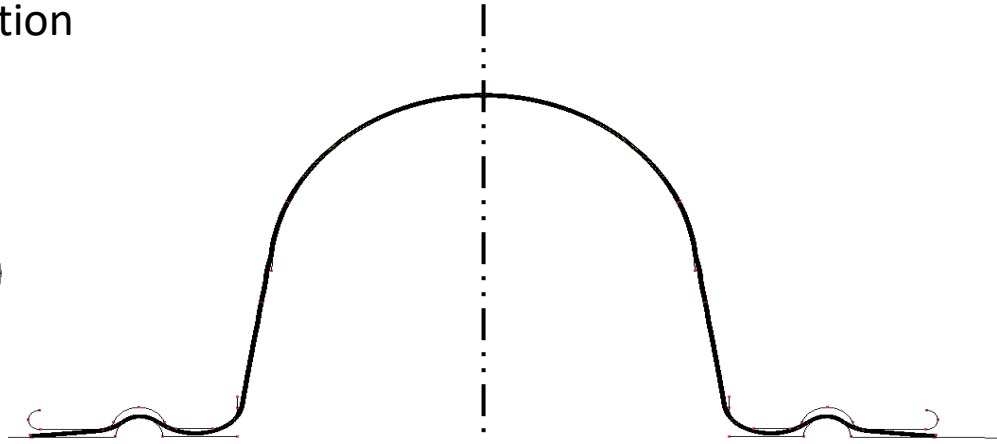
1. Each time you start a new simulation, **you MUST restart Metafor**! (unfortunately, PhD students and researchers do not free the memory they use in their routines…)

2. Some other interesting examples from the `apps` folder (among many others!)

- `apps.qs.ringtest` : a tribological test
- `apps.iso.amor` : a shock absorber
- `apps.qs.ddrawing` : deep drawing simulation
- `apps.qs.tube` : hydroforming (3D)
- `apps.qs.nine` : drawbead simulation



`apps.qs.ringtest`



`apps.qs.ddrawing`

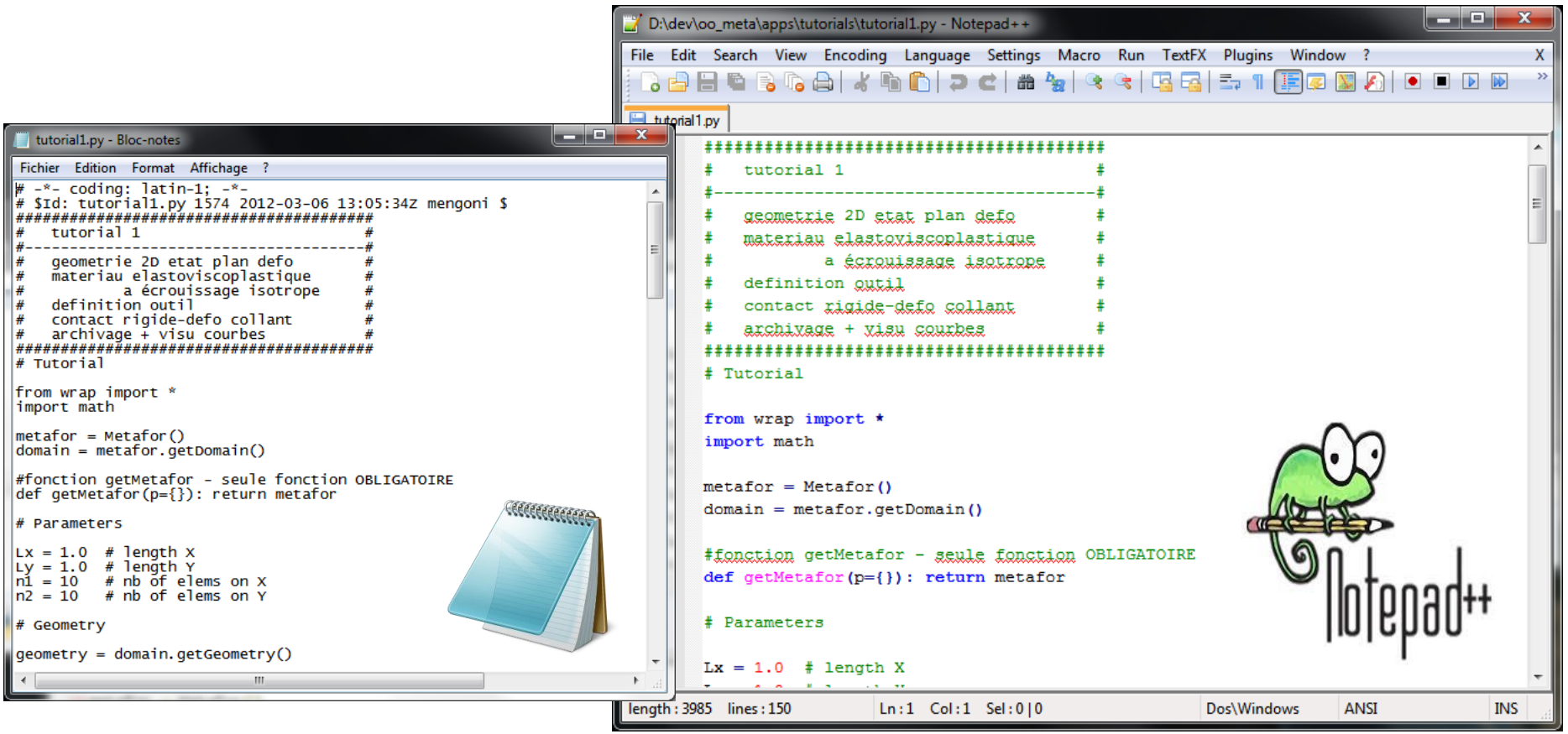
Outline

1. What is Metafor?
2. How to install Metafor?
3. How to run an existing test?
4. **How to modify an existing test?**
5. FAQ

How to modify an existing test?

Which text Editor?

Although Windows NotePad can be used to write/modify python code, it is a very good idea to install a more developer-oriented text editor (NotePad++, Atom, Visual Studio Code, etc. – extensive list here).

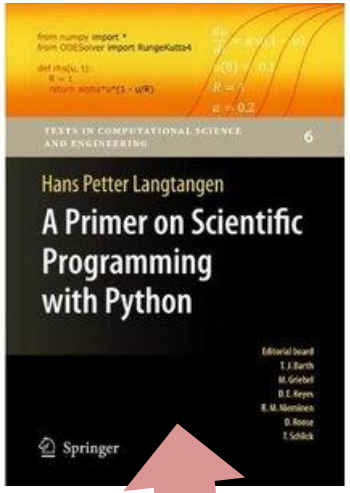


How to modify an existing test?

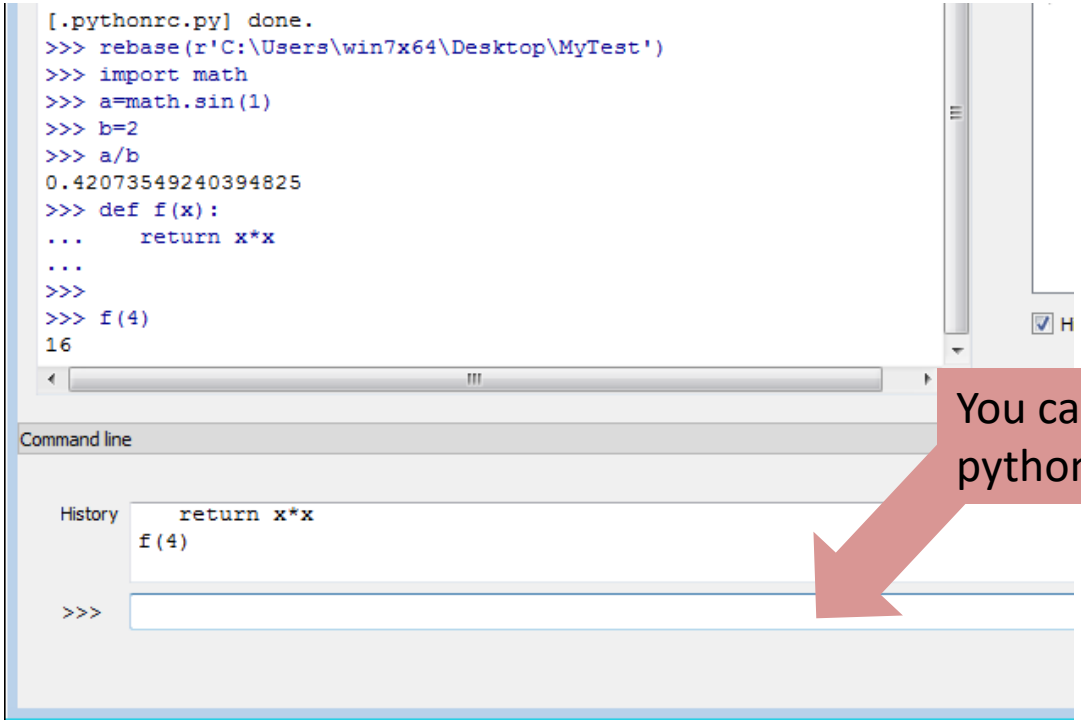
First, learn (or remember) the basics of **Python 2.x**

- Quick summary of python on Metafor website
- Python official tutorial
- ebooks

(Learning python is not a waste of time for engineers)



Lots of ebooks are freely (and legally) available from the ULg Library website!



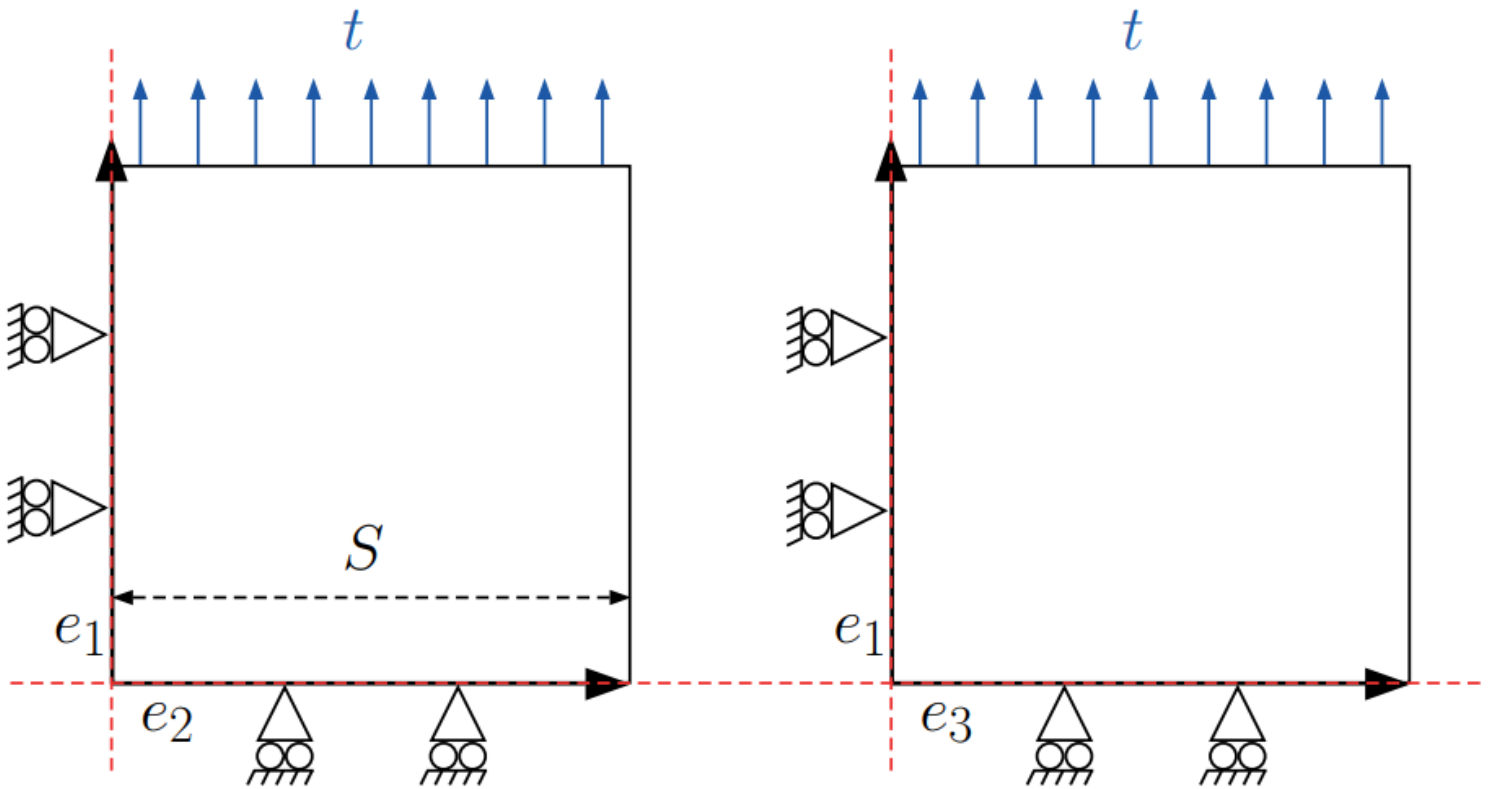
You can use Metafor as a classical python interpreter

python™

How to modify an existing test?

The python input file related to this project is** CubeSurfaceTraction.py**:

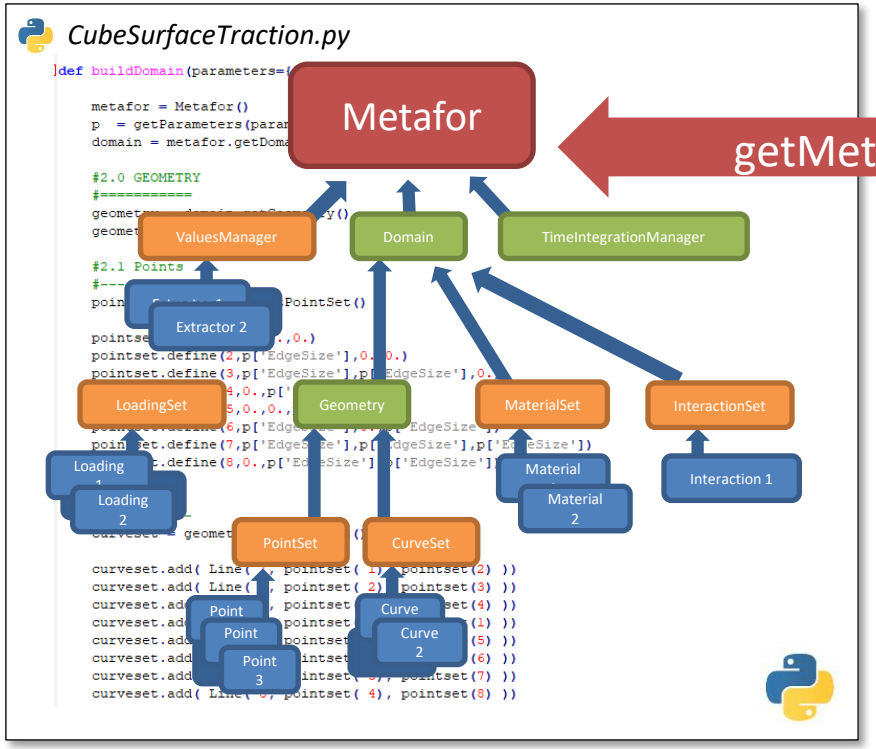
Simple test of a cube subjected to surface traction loading/unloading

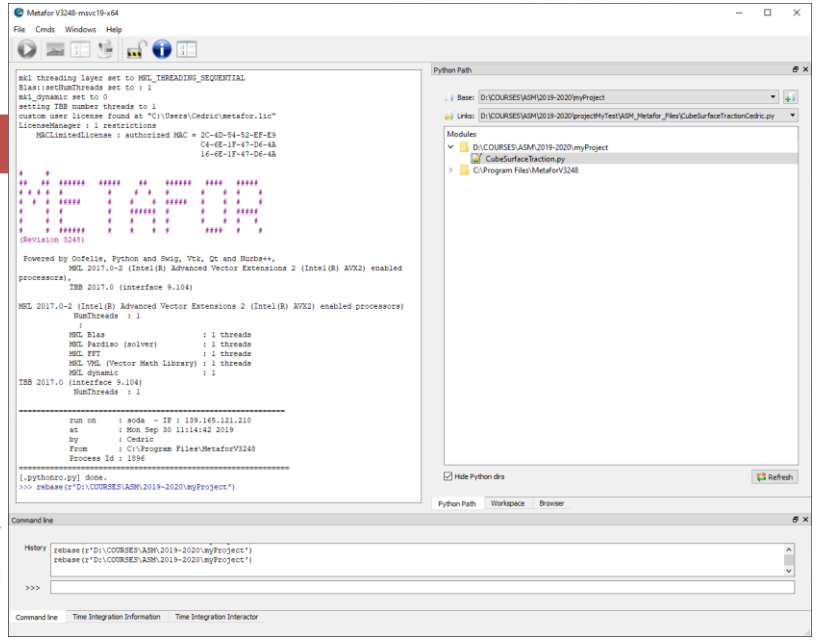


How to modify an existing test?

What is written in the file CubeSurfaceTraction.py ?

- An input file of Metafor is a python module which contains a function `getMetafor(p)`
- This function is supposed to create a `Metafor` object
- The `Metafor` object is the main object containing **everything** about a FE simulation (geometry, mesh, boundary conditions, materials, integration scheme, etc.)

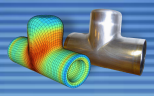




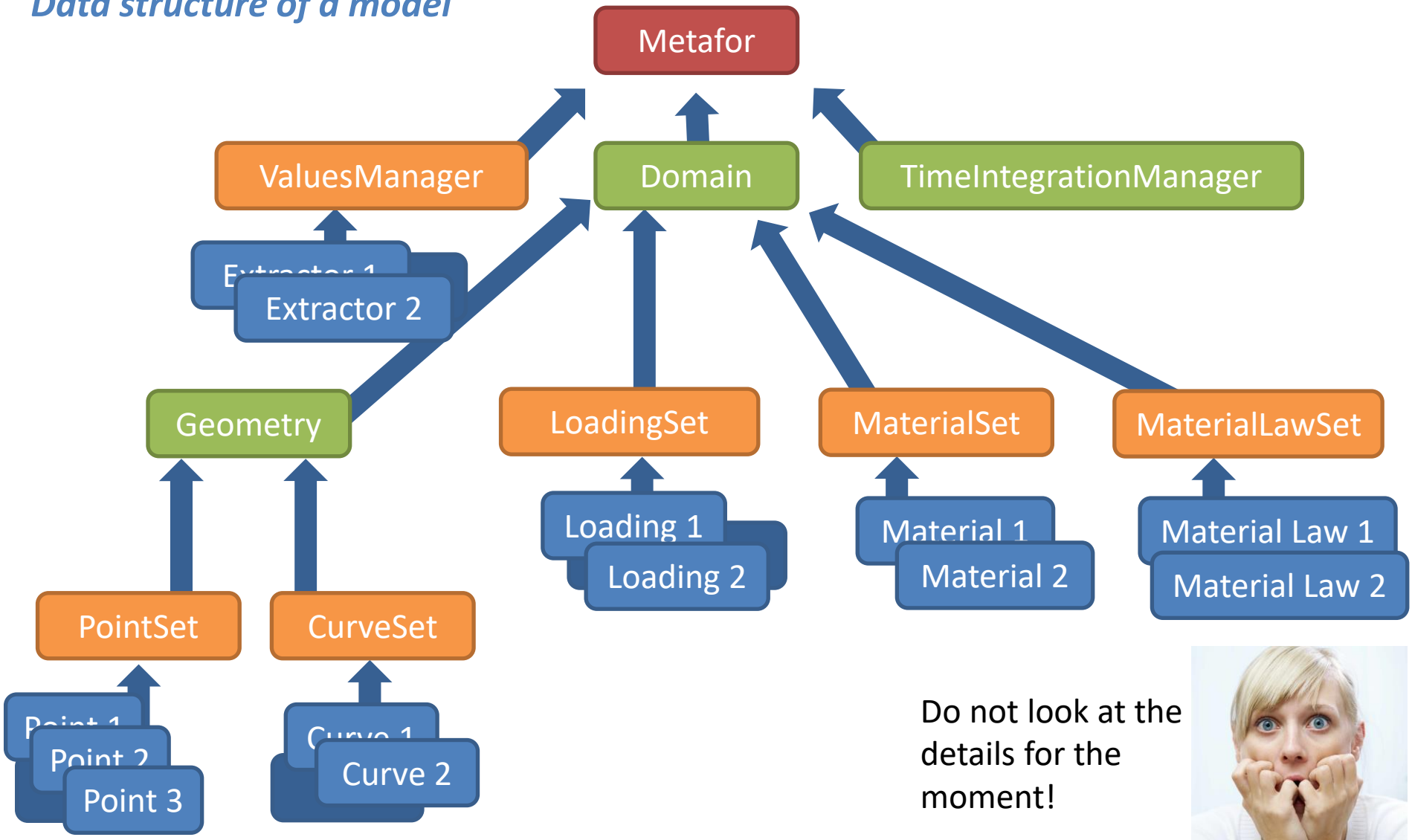
Main idea:

The python input file creates a complete data structure and returns it to the FE code.

How to modify an existing test?

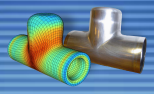


Data structure of a model



Do not look at the details for the moment!

How to modify an existing test?



Data structure of a model

Large Deformation of Solids Course
MECA-0464

LoadingSet

Loading 1

Loading 2

MaterialSet

Material 1

Material 2

MaterialLawSet

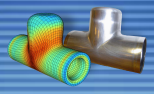
Material Law 1

Material Law 2

In this course:
Only Materials and Loadings

Interested in more?
→ Large Deformation of Solids Course MECA-0464

How to modify an existing test?



Definition of several parameters through python variables

```
#GEOMETRY:
p= {}
p['GeometryHypothesis'] = "PLANESTRAIN"
#PLANESTRESS or "PLANESTRAIN"

p['EdgeSize'] = 100 #Length of the cube

#MESH:
p['Nx'] = 1 #Nb of elements in the x direc
p['Ny'] = 1 #Nb of elements in the y direction
p['Nz'] = 1 #Nb of elements in the z direction

#TIME:
p['dT'] = 0.1 #Maximum time step
```

It is **very important** to define parameters through python variables

The model will be **parameterized**

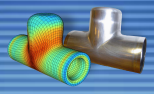
Changing the length of the cube will only require the modification of one single line of the header of the file

Here a python dictionary “p{}” is used to differentiate them easily.

```
#GEOMETRY:
GeometryHypothesis = "PLANESTRAIN"
EdgeSize = 100
#MESH:
Nx = 1
Ny = 1
Nz = 1
#TIME:
dT = 0.1
```

But python variables can also be defined directly like this

How to modify an existing test?



Geometry of CubeSurfaceTraction.py: DO NOT MODIFY

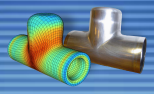
```
#2.0 GEOMETRY
#============
geometry = domain.getGeometry()
geometry.setDim3D()

#2.1 Points
#----------
pointset = geometry.getPointSet()
pointset.define(1,0.,0.,0.)
pointset.define(2,p['EdgeSize'],0.,0.)
pointset.define(3,p['EdgeSize'],p['EdgeSize'],0.)
pointset.define(4,0.,p['EdgeSize'],0.)
pointset.define(5,0.,0.,p['EdgeSize'])
pointset.define(6,p['EdgeSize'],0.,p['EdgeSize'])
pointset.define(7,p['EdgeSize'],p['EdgeSize'],p['EdgeSize'])
pointset.define(8,0.,p['EdgeSize'],p['EdgeSize'])

#2.2 Curves
#-----------
curveset = geometry.getCurveSet()
curveset.add( Line( 1, pointset( 1), pointset(2) ))
curveset.add( Line( 2, pointset( 2), pointset(3) ))
curveset.add( Line( 3, pointset( 3), pointset(4) ))
curveset.add( Line( 4, pointset( 4), pointset(1) ))
curveset.add( Line( 5, pointset( 1), pointset(5) ))
curveset.add( Line( 6, pointset( 2), pointset(6) ))
curveset.add( Line( 7, pointset( 3), pointset(7) ))
curveset.add( Line( 8, pointset( 4), pointset(8) ))
curveset.add( Line( 9, pointset( 5), pointset(6) ))
curveset.add( Line( 10, pointset( 6), pointset(7) ))
curveset.add( Line( 11, pointset( 7), pointset(8) ))
curveset.add( Line( 12, pointset( 8), pointset(5) ))
```

Access the geometry and set hypothesis

Create Points

Create Curves

How to modify an existing test?



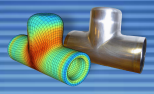
***Geometry of** CubeSurfaceTraction.**py**:* **DO NOT MODIFY**

```
#2.3 Wires
#------------
wireset = geometry.getWireSet()
wireset.add( Wire(1, [curveset(1), curveset(2), curveset(3), curveset(4)]) )
wireset.add( Wire(2, [curveset(9), curveset(10), curveset(11), curveset(12)]) )
wireset.add( Wire(3, [curveset(1), curveset(6), curveset(9), curveset(5)]) )
wireset.add( Wire(4, [curveset(2), curveset(7), curveset(10), curveset(6)]) )
wireset.add( Wire(5, [curveset(3), curveset(7), curveset(11), curveset(8)]) )
wireset.add( Wire(6, [curveset(4), curveset(8), curveset(12), curveset(5)]) )

#2.4 Sides
#-------------------
sideset = geometry.getSideSet()
sideset.add( Side(1,[wireset(1)]) )
sideset.add( Side(2,[wireset(2)]) )
sideset.add( Side(3,[wireset(3)]) )
sideset.add( Side(4,[wireset(4)]) )
sideset.add( Side(5,[wireset(5)]) )
sideset.add( Side(6,[wireset(6)]) )

#2.5 Skins
#-------------------
skinset = geometry.getSkinSet()
skinset.add(Skin(1,[sideset(1),sideset(2),sideset(3),sideset(4),sideset(5),sideset(6)]))

#2.6 Volume
#-------------------
volumeset = geometry.getVolumeSet()
volumeset.add(Volume(1,[skinset(1)]))
```

Create Sides

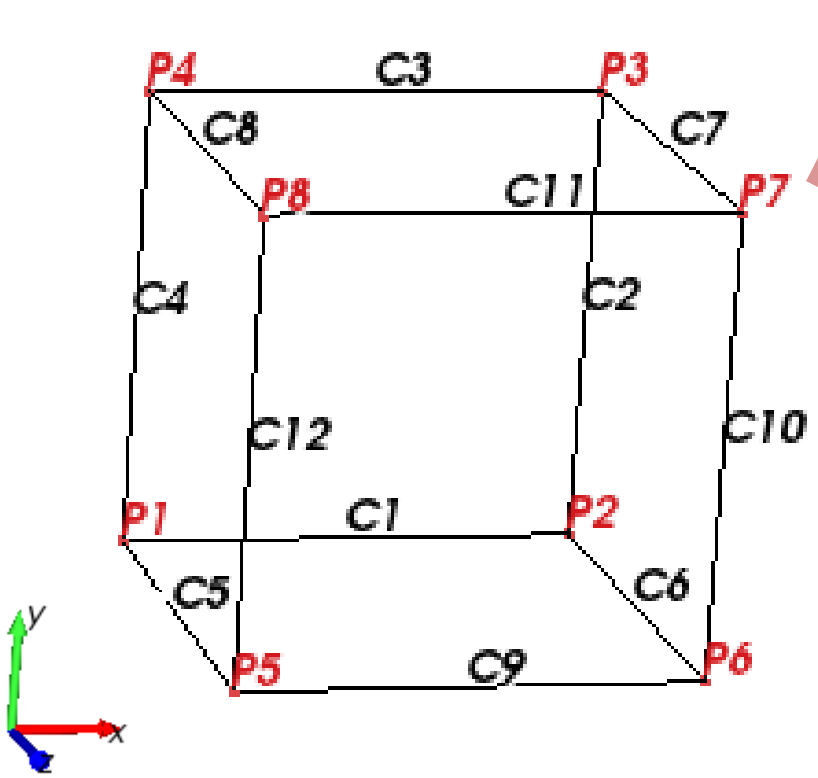
Create Volume

How to modify an existing test?



Geometry of** CubeSurfaceTraction.**py:

Resulting geometry:



Point label

Curve label

Side label:

S1: C1→C2 →C3 →C4

S2: C9→C10→C11→C12

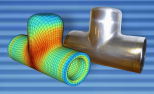
S3: C1→ C6 → C9 → C5

S4: C2→C7 →C10 →C6

S5: C3→C7 →C11 →C8

S6: C4→C8 →C12 →C5

How to modify an existing test?



Definition of the constitutive behaviour of the solid

```
Density = 7.00E-9 #Density
Young = 20.5E4 #Young's Modulus
Nu = 0.33 #Poisson ratio

materset = domain.getMaterialSet()
material1 = materset.define (1, EvpIsoHHypoMaterial)
material1.put(MASS_DENSITY, Density)
material1.put(ELASTIC_MODULUS, Young)
material1.put(POISSON_RATIO, Nu)
material1.put(YIELD_NUM,1)
```

Materials are stored in the `MaterialSet` of the `Domain`

The material number is 1. `EvpIsoHHypoMaterial` is the material type.

See the documentation for a complete list of materials

Material parameters are defined with a series of `put` commands such as

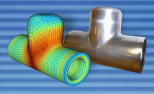
`material.put(MATERIAL_CODE, value)`

See the documentation for a complete list of available parameters for each material

! Units must be consistent !

If lengths in [mm], then
- forces in [N],
- stresses in [N/mm²] = [MPa]
- density in [T/mm³]

How to modify an existing test?



Definition of the constitutive behaviour of the solid

```
material1.put(YIELD_NUM,1)

SigmaY_0=200.0 #Elastic limit of virgin material
h = 16000.0 #Hardening parameter

lawset = domain.getMaterialLawSet()
lawset1 = lawset.define(1, LinearIsotropicHardening)
lawset1.put(IH_SIGEL, SigmaY_0)
lawset1.put(IH_H, h)
```

The material requires a hardening law (we have assigned law#1 which does not exist yet)

Material laws such as hardening laws are stored in the `MaterialLawSet` of `domain`.

The syntax is the same as for `MaterialSet`.

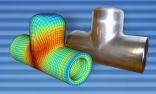
See the documentation for an comprehensive list of available laws

! Units must be consistent !

If lengths in [mm], then

- forces in [N],
- stresses in [N/mm²] = [MPa]
- density in [T/mm³]

How to modify an existing test?



Definition of the properties of the finite elements

```
prp1 = ElementProperties (Volume3DElement)
prp1.put (MATERIAL, 1) # Number of the material used
prp1.put (CAUCHYMECHVOLINTMETH,VES_CMVIM_STD)
```

Set element type

Set integration method

Assign material #1 (Defined previously)

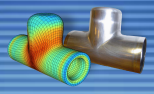
You can create multiple materials in one file and switch them here.

Or

Create 1 input file per new material.

DO NOT MODIFY

How to modify an existing test?

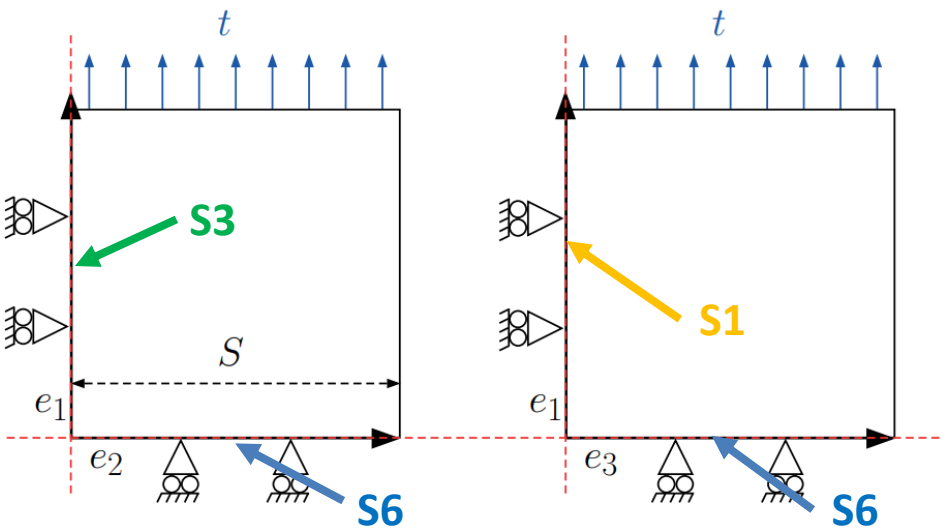


***Boundary conditions in** CubeSurfaceTraction.**py**:*

Loads and prescribed d.o.f.s are defined in the `LoadingSet` of `domain`

Prescribed value

```
loadingset = domain.getLoadingSet()

if p['GeometryHypothesis']=="PLANESTRESS":
        loadingset.define(sideset(1),Field1D(TZ,RE),0.)
        loadingset.define(sideset(3),Field1D(TY,RE),0.)
        loadingset.define(sideset(6),Field1D(TX,RE),0.)
```



Field code:

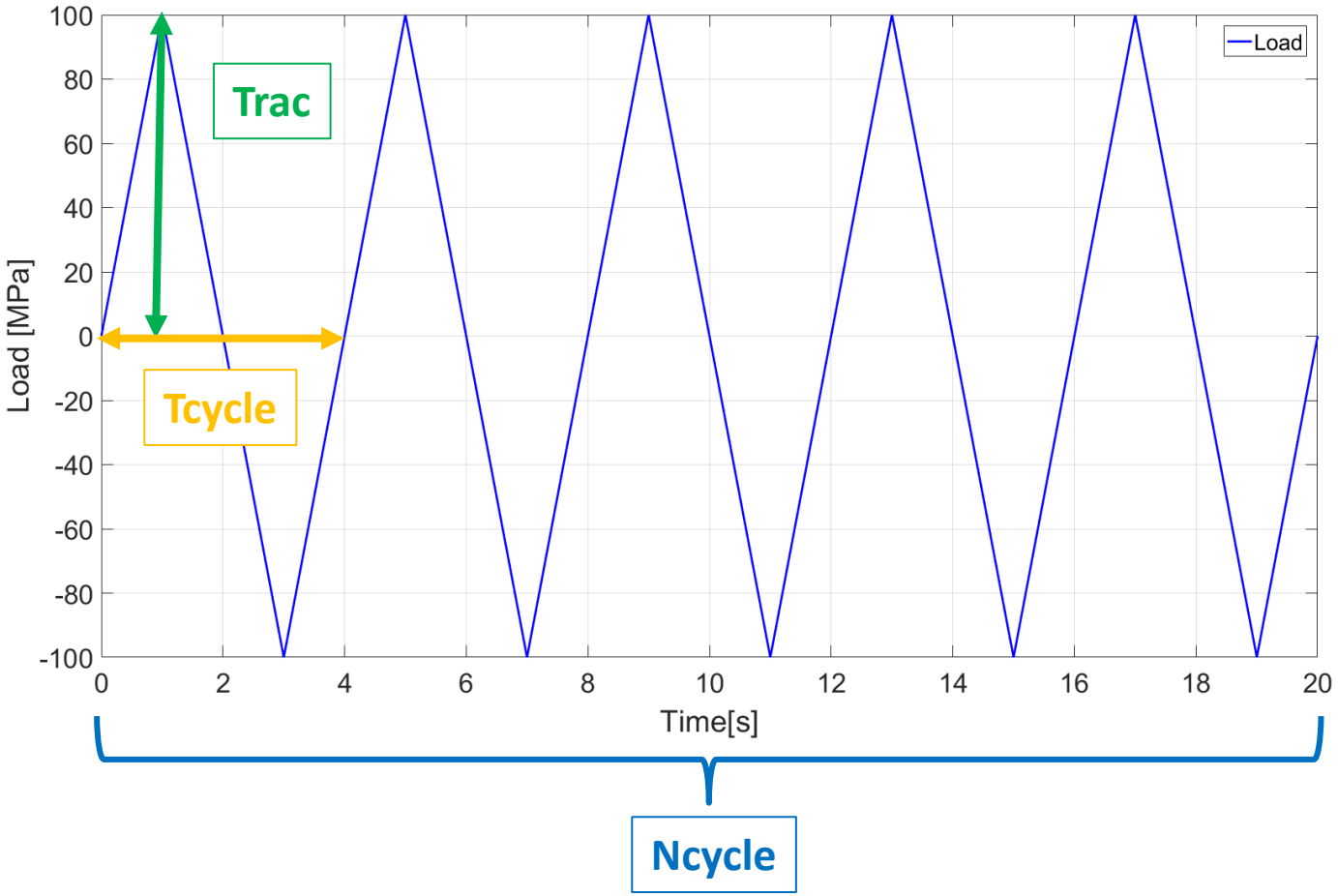
- `Field1D(TX,RE)`: x displacement
- `Field1D(TY,RE)`: y displacement
- `Field1D(TZ,RE)`: z displacement

See documentation for a full list of codes.

How to modify an existing test?

***Load in** CubeSurfaceTraction.**py**:*

```
#LOAD:
Trac = 100  #Traction
Ncycle = 5  #Number of cycles of loading/unloading
Tcycle = 4. #Duration of one cycle
```



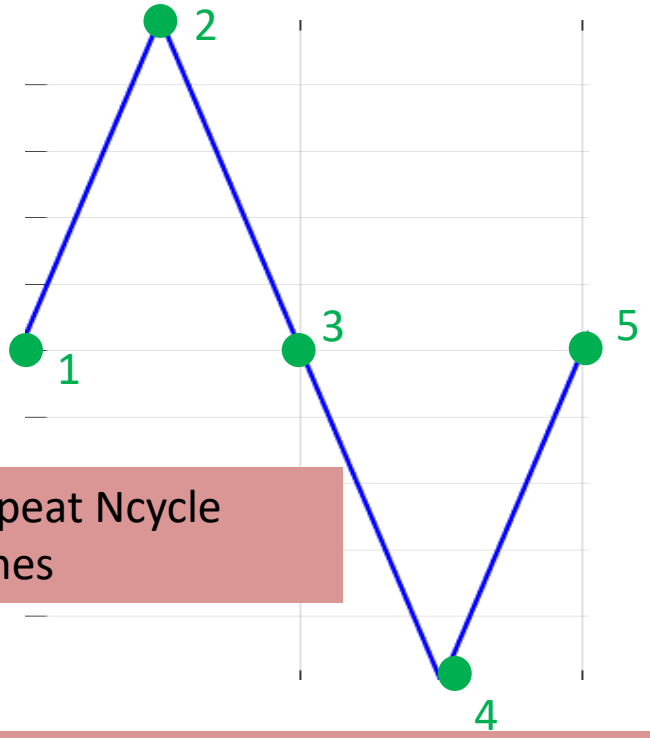
How to modify an existing test?

***Load in** CubeSurfaceTraction.**py**:*

`PieceWiseLinearFunction` is a piecewise-linear dependance on time. Each point is created via `fct.setData(time, value)`

```
fct = PieceWiseLinearFunction()
[redacted]
      fct.setData([redacted]0.,0.) #POINT n°1
      fct.setData([redacted]Tcycle/4.0,1.0)#POINT n°2
      fct.setData([redacted]Tcycle*2.0/4.0,0.0)#POINT n°3
      fct.setData([redacted]Tcycle*3.0/4.0,-1.0)#POINT n°4
      fct.setData([redacted]Tcycle*4.0/4.0,0.0)#POINT n°5
```

Repeat Ncycle times



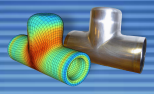

```
prp2 = ElementProperties (Traction3DElement)
prp2.put(PRESSURE, Trac)
prp2.depend (PRESSURE, fct, Field1D(TM,RE))
```

Prescribed Pressure at time `t` is `Trac(t) = Trac*fct(t)`

```
trac = LoadingInteraction(2)
trac.push(sideset(4))
trac.addProperty(prp2)
domain.getInteractionSet().add(trac)
```

The load is applied perpendicular to side n°4

How to modify an existing test?



History curves – extraction of a value at each time step

History curves are also called `ValueExtractors`. They are stored in the `ValuesManager` of `Metafor`.

Writes the value of the time at each step in the file `time.ascii`

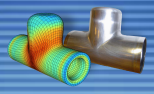
```
valuesmanager = metafor.getValuesManager()

valuesmanager.add(1, MiscValueExtractor(metafor,EXT_T),'time')

valuesmanager.add(2, IFNodalValueExtractor(pointset(node_id), IF_SIG_XX),'Sigma_XX' )
valuesmanager.add(3, IFNodalValueExtractor(pointset(node_id), IF_SIG_YY),'Sigma_YY' )
valuesmanager.add(4, IFNodalValueExtractor(pointset(node_id), IF_SIG_ZZ),'Sigma_ZZ' )
valuesmanager.add(5, IFNodalValueExtractor(pointset(node_id), IF_EVMS),'SigmaVM' )
```

The next extractor writes the XX component of the nodal internal stress at node ‘node_id ’ and saves it to a file named `Sigma_XX.ascii`.

The same is done for `Sigma_YY`, `Sigma_ZZ` and `SigmaVM`

Lots of extractors are available… see documentation

How to modify an existing test?



History curves – extraction of a value at each time step

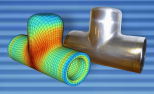
The given code CubeSurfaceTraction.py **already extracts:**

- **Stresses:** SigmaXX, SigmaYY, SigmaZZ, SigmaVM
- **Strains:** E_XX, E_YY, E_ZZ, E_PL(equivalent plastic strain)
- **BackStress:** A_XX, A_YY, A_ZZ

All extracted at node 7.

Those curves are sufficient to carry out the project but you are allowed to extract others if you deem it usefull.

How to modify an existing test?

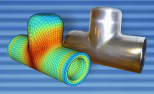


History curves – visualization

```
#Stress
dataCurveSX = VectorDataCurve(1, valuesmanager.getDataVector(1), valuesmanager.getDataVector(2),'Sigma_XX')
dataCurveSY = VectorDataCurve(2, valuesmanager.getDataVector(1), valuesmanager.getDataVector(3),'Sigma_YY')
dataCurveSZ = VectorDataCurve(3, valuesmanager.getDataVector(1), valuesmanager.getDataVector(4),'Sigma_ZZ')
dataCurveSVM = VectorDataCurve(4, valuesmanager.getDataVector(1), valuesmanager.getDataVector(5),'Sigma_VM')

dataCurveSet2 = DataCurveSet()
dataCurveSet2.add(dataCurveSX)
dataCurveSet2.add(dataCurveSY)
dataCurveSet2.add(dataCurveSZ)
dataCurveSet2.add(dataCurveSVM)

winc2 = VizWin()
winc2.add(dataCurveSet2)
metafor.addObserver(winc2)
```

This piece of code can be used to add a second graphical windows which displays a 2D plot of selected curves during the simulation.

How to modify an existing test?



How to easily switch from materials/ geometry?

Option 1:
Use parameters combined with an if function

```
if p['GeometryHypothesis']=="PLANESTRESS":
        loadingset.define(sideset(1),Field1D(TZ,RE),0.)
        loadingset.define(sideset(3),Field1D(TY,RE),0.)
        loadingset.define(sideset(6),Field1D(TX,RE),0.)

elif p['GeometryHypothesis']=="PLANESTRAIN":
        loadingset.define(sideset(1),Field1D(TZ,RE),0.)
        loadingset.define(sideset(3),Field1D(TY,RE),0.)
        loadingset.define(sideset(6),Field1D(TX,RE),0.)
        # IMPLEMENT BOUNDARY CONDITION TO OBTAIN PLANE STRAIN STATE HERE #
```

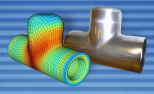
Option 2:
Use comments and/or create new .py files

```
loadingset = domain.getLoadingSet()
loadingset.define(sideset(1),Field1D(TZ,RE),0.)
loadingset.define(sideset(3),Field1D(TY,RE),0.)
loadingset.define(sideset(6),Field1D(TX,RE),0.)
#Uncomment to use plane strain:
#loadingset.define(sideset(1),Field1D(TZ,RE),0.)
#loadingset.define(sideset(3),Field1D(TY,RE),0.)
#loadingset.define(sideset(6),Field1D(TX,RE),0.)
## IMPLEMENT BOUNDARY CONDITION TO OBTAIN PLANE STRAIN STATE HERE #
```

Outline

1. What is Metafor?
2. How to install Metafor?
3. How to run an existing test?
4. How to modify an existing test?
5. **FAQ**

FAQ



I still don't know how to…?

In this order:

1) Read again this presentation (**everything is here**).

2) Read the documentation (use google search and the search box on the web site).

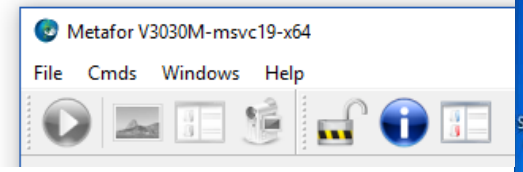
3) Ask your question at the Q/A sessions or on the forum.

4) Send a mail to cedric.laruelle@uliege.be

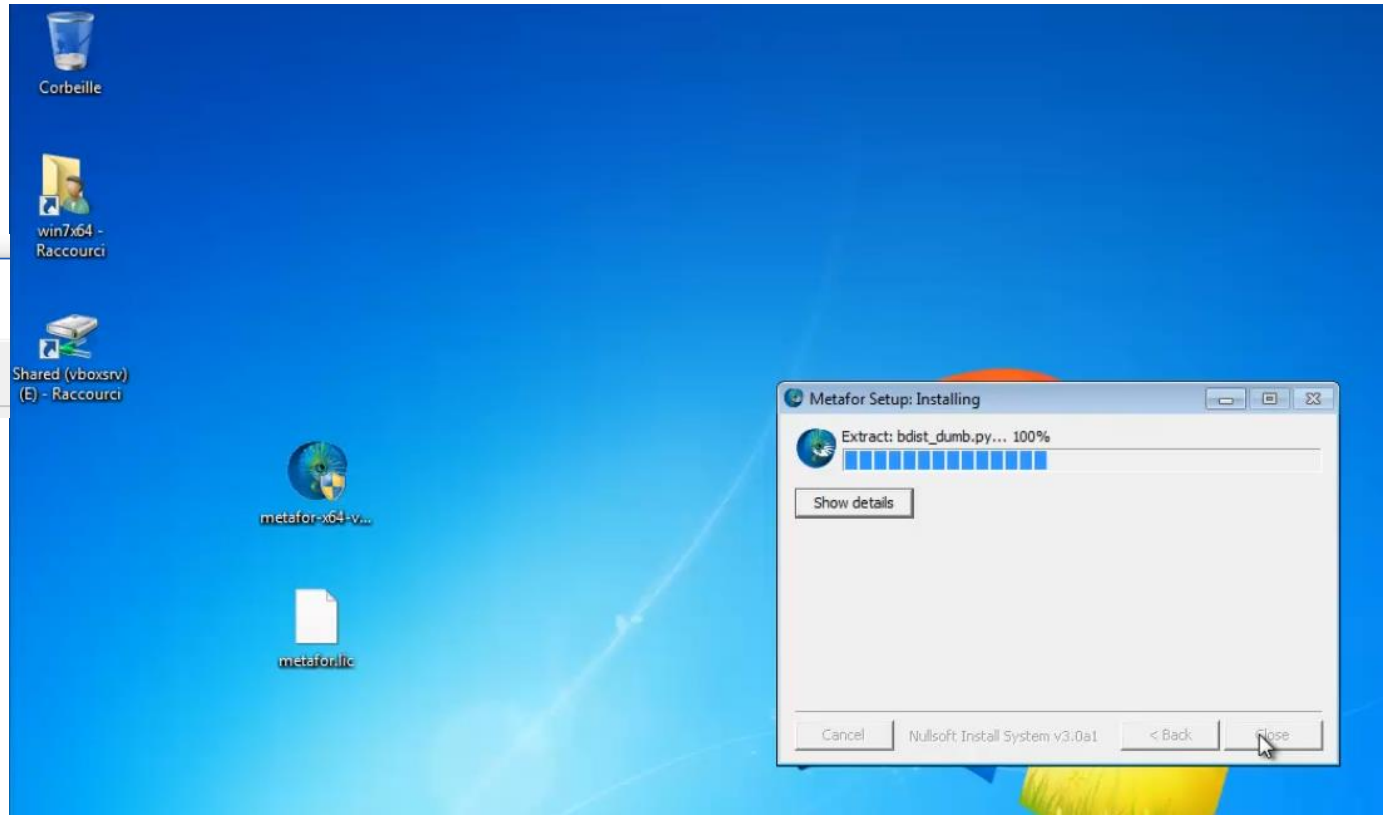
I want to try larger tests

Procedure: license installation (« 5000 nodes for 1 year » instead of « 500 forever »)

- Download the license from metafor.ltas.ulg.ac.be
- Run Metafor (desktop link)
- Click on the padlock in order to import the license file (`metafor.lic`)
- Restart Metafor



import license



See also:

- Youtube video
- Documentation (slightly outdated)